AFRL-HE-WP-TR-1999-0015

UNITED STATES AIR FORCE RESEARCH LABORATORY

Technical Report for the Integrated Requirements Support System (IRSS)

Mark Yovish

Donald Padgett

Julie Mann

BOOZ-ALLEN & HAMILTON, INC.
8283 Greensboro Drive
McLean VA 22102-3838

January 1999

Final Report for the Period February 1998 to November 1998

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Deployment and Sustainment Division
Sustainment Logistics Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner, licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield VA 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., STE 0944
Ft Belvoir VA 22060-6218

DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-1999-0015

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

For Lt Col

JAY KIDNEY, Lt Col, USAF, Chief
Deployment and Sustainment Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1999	Final - February 1998 - November 1998

4. TITLE AND SUBTITLE

Technical Report for the Integrated Requirements Support System (IRSS)

5. FUNDING NUMBERS

C - F41624-98-F-5014
PE - 63106F
PR - 2745
TA - 03
WU - 06

6. AUTHOR(S)

Mark Yovish; Donald Padgett; Julie Mann

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Booz-Allen & Hamilton Inc.
8283 Greensboro Drive
McLean, VA 22102-3838

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, Human Effectiveness Directorate
Deployment and Sustainment Division
Air Force Material Command
Sustainment Logistics Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-HE-WP-TR-1999-0015

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Air Force is seeking technology that will facilitate the requirements component of the Air Force Acquisition process. Collaboration in action is much more than the application of a technology, it is about transforming coordination, a relatively labor intensive activity, into collaboration--a rich and rigorous discussion forum that optimizes the decision making environment. The Integrated Requirement Support System (IRSS) applies technology to the requirement process through a personal computer software application of core functions. It would assist functional experts in developing, tracking, coordinating, and staffing key modernization planning documents (such as mission need statements, capstone requirements documents and operational requirements documents). Many functional communities contribute information to and draw information from these documents. Each community has its own focus; e.g., tactics and weapon system deployment, funding and contracting, environmental compatibility, supportability, human factors engineering, system safety, etc. It should be noted that the intent of the IRSS was not to enforce a single pattern of information across all organizations, but to permit sharing of information across unique requirement domains.

14. SUBJECT TERMS

IRSS; Requirements; requirements specification; functionality; coordination; colloboration; functional contracts; database environment; client-server architecture; communication protocols; project management information; ORD; MNS; POM; information sharing

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This report documents the results of a technological effort to provide the Air Force Requirements Community an application that would support core requirements generation. This effort was part of a logistics research and development program title Integrated Requirements Support System (IRSS) (Contract Number F41624-98-F-5014) managed by the Air Force Research Laboratory, Logistics Sustainment Branch (AFRL/HESS), at Wright-Patterson AFB, OH. The advanced development (6.3) research produced a proof-of-concept, the Integrated Requirements Support System (IRSS) which will be transitioned for further development for the Air Force requirements process participants thereby enabling to continue the process improvement effort.

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

1 Introduction

The Air Force Research Laboratory, Sustainment Logistics Division (AFRL/HESS) is conducting research and development to support engineering, technical and change management for the requirements component of Air Force Acquisition process. In support of this process, Major Commands (MAJCOMs) and Field Operating Agencies (FOAs) develop requirements that eventually evolve into acquisition programs. The Integrated Requirements Support System (IRSS) is an AF advanced development research program championed by the AF Director of Operational Requirements (HQ USAF/XOR). The Air Force Research Laboratory sponsors IRSS research, development and testing. The principals of the Air Force Requirements Oversight Council have endorsed the definition and test of automated support for a virtual AF requirements process. Organizations participate in IRSS through the IRSS Integrated Product Team. AFRL/HESS tasked Booz·Allen & Hamilton to develop and implement an IRSS prototype at a number of Air Force field sites.

AFRL/HESS has instituted an IPT to participate in the requirements definition, design, implementation and testing of IRSS. The IPT currently consists of a representative from the following organizations:

- Air Combat Command (ACC)
- Air Force Special Operations Command (AFSOC)
- Air Force Space Command (AFSPC)
- Air Mobility Command (AMC)
- Air Education and Training Command (AETC)
- Air Force Operational Test and Evaluation Center (AFOTEC)
- Air Intelligence Agency (AIA)
- Air Force Research Laboratory (AFRL/HESS)
- Space and Missile Center (SMC)
- Air Staff (AF/XORD)

IPT members participate in periodic IRSS Program Reviews, provide functional and design inputs to the development of IRSS, and serve as members of the Configuration Control Board (CCB). Member organizations comprise the field sites for IRSS implementation.

1.1 IRSS Definition

The objective of IRSS is to demonstrate the benefits of an Air Force-wide, user-driven and user-supported, integrated requirements system. The IRSS testbed is intended to facilitate and integrate the operational requirements definition, coordination, and management activities of the warfighting commands, the Air Staff, and partners such as AF Operational Test and Evaluation Center (AFOTEC), and AF Intelligence Agency (AIA) and provide a single injection point for requirements engineering innovations.

IRSS uses a synchronized client/server database technology in a client-based processing environment. The IRSS client is developed in PowerBuilder 6.0 and interacts with an Oracle database. Oracle is the most powerful and capable Relational Database Management System (RDBMS) available and PowerBuilder is the industry leader for client/server graphical user interface development. The client operates on Windows 95 or NT 32-bit operating system while the servers operate under Windows NT or any Oracle compatible operating system. All clients and servers communicate via the TCP/IP protocol. The IRSS technical architecture, founded on PowerBuilder's object oriented principles and Oracle Workgroup Server are further explained throughout this document. The current IRSS architecture employs nine unclassified IRSS servers (HQ USAF, ACC, AFSOC, AETC, AIA, AFOTEC, AFSPACE, AMC, AFMC), and one classified server (Space Warfare Center). Booz·Allen is the designer, developer, and maintainer of IRSS.

The client/server environment allows the easy exchange, coordination and collaboration of Air Force Requirements information. IRSS is designed for use by both action officers and the senior leadership of an organization. IRSS will support action officers in the development, validation and fulfillment of operational needs or requirements necessary for satisfying shortfalls or improving military capabilities. The IRSS functionality is based on a project. To support daily action officer activity, IRSS captures the people, tasks and documents associated with a project. Traditionally, the project is the subject of a Mission Needs Statement (MNS) or Operational Requirements Document (ORD), but a project is anything people work on. In IRSS, a project is defined by the user and can include items such as a Mission Area Plan (MAP) deficiency, a Command need or a future concept. To a project, the users relate the people working on the project, the tasks needed to complete the project, and the documents generated as part of the project. IRSS also provides senior leadership with executive-level synopses of the work in progress within their organization to meet Air Force requirements. A more detailed explanation of IRSS functionality is described in Section 3.

Additionally, IRSS captures the project justification, i.e., deficiencies contained in the MAPs, and has the ability to capture and process the budgetary programming POM data related to the project. A fundamental value of IRSS is its ability to make the entire universe of requirements data available in a corporate environment. The information helps the requirements community see the progression of a specific project, it helps decision makers see the Science & Technology community's response to deficiencies, it highlights MAJCOM efforts on documents, and identifies duplication and overlapping of deficiency resolutions.

IRSS also provides a variety of supporting functional capabilities including:

- Storing requirements electronically including:
  - People involved
  - Supporting documentation
  - Tracking, tracing, and status.
- Developing and preparing documentation including:
  - Building initial documents
  - Coordination
  - Collecting, resolving, and archiving comments.
- Summarizing requirements including:

- Key management metrics (e.g., summaries, timelines, reports)
- Supporting data.

IRSS is also being evaluated by the AF planning community (AF/XPX) as a collaboration tool for the development of AF plans such as the Mission Area Plans and the AMC Master Plan. The eventual goal is to have a seamless flow of information from the requirements community to the planning community and then into the POM via the programming community. As IRSS is quickly evolving from an AFRL R&D project to the foundation of the AF requirements process, the AFROC has tasked AF/XOR to create a program office for IRSS.

IRSS is in use today and the MAJCOMs are in the process of migrating both data and business processes from legacy systems to IRSS. For the initial one year effort, AFRL tasked Booz·Allen to implement IRSS in three prototypes, each building on the capabilities and functionality of the previous. IRSS is currently in its second year of development and AFRL has tasked Booz·Allen to continue the maintenance and development of IRSS in two additional prototypes. This document will subsequently refer to the IRSS prototypes as versions.

2 Approach to IRSS Development

Booz·Allen's approach to designing and developing IRSS employs the principles of Rapid Application Development (RAD) and the extensive use of Joint Application Design (JAD) sessions. This methodology ensures user requirements are quickly incorporated into prototype applications then released to users for functionality and ease of use validation. RAD allows Booz·Allen to develop IRSS incrementally, enhancing and changing the application and server architecture as we move through the versions. This approach is critical for an application with a wide variety of users in distributed locations. The RAD methodology requires continual user feedback and critique and IRSS development supports this approach. Booz·Allen is also releasing interim versions (e.g., 1.01, 3.1) to field sites to ensure IRSS meets the needs of all users by quickly correcting application errors.

2.1 IRSS Requirements Analysis

This section presents an overview of the methodology used to gather and validate IRSS functional requirements and summarizes the results of the analysis in terms of major functional areas, called computer software configurable items (CSCIs). The purpose of the requirements analysis was twofold:

- Consolidate each IRSS field site requirement into a common set of functional and system requirements
- Incorporate those requirements in the design and implementation of IRSS.

2.1.1 Requirements Analysis Methodology

The IPT and IRSS program office provide Booz·Allen with a list of user requirements generated from a group decision support exercise (Appendix A). These initial requirements comprised the functionality required for IRSS implementation in three versions. Booz·Allen used these requirements as a baseline and supplemented the approach to gathering, assimilating, and validating IRSS functional and system requirements through the following activities:

- IRSS IPT Discussions
- Conduct interviews with AF functional experts
- Reverse engineer existing Requirements Tracking System
- AFI 10-601 and DoD 5000.2
- Joint Application Design (JAD) sessions.

The results of the interviews and reverse engineering exercises were captured in ERwin, a Computer Automated Software Engineering (CASE) tool, in the form of entity relationship diagrams. The diagrams graphically define the relationships of specific entities involved in the

Air Force requirements process. Booz·Allen used ERwin to produce entity relationship models, compile a data dictionary, and list narrative descriptions of the entities and relationships.

2.1.2 Requirements Analysis Findings

As a result of requirements gathering and validation, Booz·Allen developed an IRSS system concept that describes required system functionality. The IRSS system concept classified the user requirements into four major functional CSCIs. The CSCIs include:

- *People*—People, in the requirements process, generate and develop projects and documents and inform others of the requirement's progress. The activities of People in processing a project from its beginning through procurement must be captured by IRSS. Likewise, IRSS must enable the sharing of data among authorized users.
- *Documents*—A Document, in the requirements process, contains the data and information needed to support a requirement as it moves through the requirements process. IRSS will assist in developing documents as well as tracking documents throughout this process.
- *Projects*—Projects generally evolve from an inability to perform an assigned task, frequently referred to as a deficiency. For example, a requirement may begin as a deficiency or need identified in a Mission Area Plan. However, not all deficiencies or needs evolve into requirements, and not all requirements originate from a deficiency or need. As a requirement progresses into acquisition process, it goes through a series of milestones and phases, during which AF organizations perform various activities. These milestones and activities correspond with and are supported by documents.
- *System Management*—System Management defines the internal functionality relating to system performance, system security, and system administration that IRSS must perform in order to satisfy user requirements.
- *Executive Summary*—Executive Summary refers to high-level views of information contained in IRSS. It represents data in graphical and report form to facilitate the identification and exchange of important information.

2.1.3 JAD Sessions

A cornerstone of our database development process for IRSS was our use of JAD sessions/workshops, technical interchange meetings and design and progress reviews. This approach is an effective method for capturing user requirements to support the development of useful applications. JAD sessions are a proven approach to successfully defining user and system requirements. Booz·Allen's approach to database development combined traditional JAD techniques with small interactive workshops. Traditional JAD techniques, such as building and validating entity-relationship diagrams, are often too technical for participants and consume valuable time. Instead, such products were developed behind the scenes using existing documentation, select interviews, and workshop output. Then, during JAD sessions, Booz·Allen used materials that users can quickly grasp and validate, such as function diagrams and prototype screens. As appropriate, Booz·Allen met one-on-one with select experts to refine the database design.

2.1.4 IRSS Versions

Version 1 of IRSS was termed the island approach. Each participating field site had its own IRSS server installed on their Local Area Network (LAN). This configuration provided IRSS operation from client workstations attached to the LAN. When completed, IRSS Version 1 provided full visibility into requirements generation at each field site. Version 2 of IRSS provided the connectivity between participating field sites. Each field sites had visibility into the other sites based on an information sharing matrix of rules determined by the IPT members. These data sharing features are a critical component of IRSS functionality and ensures users have the capability to determine what information they want made available to the IRSS community. Version 3 (22 Dec 97) of IRSS included enhancements to IRSS functionality and fully integrated the sharing of data among field sites. Version 3.0 demonstrated the full set of functions identified by the IRSS IPT and provides a foundation for continued research and testing of classified operations, integration of other modules, and refined architecture designs. The results of IRSS acceptance testing are contained in Appendix B.

IRSS was not initially intended to be a production system. Rather, the intent is to provide the Air Force and the AFRL with a proof-of-concept implementation to verify and demonstrate a standardized, Air Force-wide approach to managing the requirements process. A major goal of IRSS is the facilitation of a common Air Force approach to requirements management. Providing connectivity and data sharing amongst the field sites will support this goal. The initial fielding of IRSS has been successful and the AFROC is now moving towards migrating IRSS to a system program office (SPO). The IRSS SPO will support IRSS interim operational capability, development of functions providing full operational capability, and efforts to interface the requirements process to the AF planning function and the AF acquisition function. Versions 4 and 5 will further enhance IRSS and provide much of the advanced research needed to prepare IRSS for the migration. Sections 3 and 4 contain functional and technical details describing the next two IRSS versions.

2.1.5 Object-Oriented Design (OOD) and Object Oriented Programming (OOP)

The IRSS client is designed using OOD principles and PowerBuilder 6.0 development software. OOD is a popular and effective way to design software, and PowerBuilder 6.0 is the leading GUI OOP development software. OOP languages typically support three major characteristics: inheritance, encapsulation and polymorphism. These characteristics allow programmers to develop isolated, black-box objects and class libraries. All programmers on the development team can use these objects and libraries. Currently, the IRSS developers have hundreds of objects and libraries available as building blocks for their coding assignments. When multiple programmers are working simultaneously and sharing source code, configuration management, version control and security become critical to guarantee success. PowerBuilder 6.0 provides the development team with the ability to secure specific source code objects and libraries, eliminating the problems inherent to programming efforts that require multiple developers coding simultaneously. Changes to the objects and libraries are deliberate, scheduled and carefully performed.

2.2 Configuration Management (CM) Process

The RAD approach, number and variety of users, multiple versions and incremental releases requires a well defined and executed plan for configuration management. The IRSS configuration management plan defines the standard practices used to implement hardware, software, and documentation configuration management in the IRSS development and operational environments. The CM process specifies procedures to uniquely identify and control the development and release of the IRSS versions. These procedures will be particularly important for IRSS as we execute an evolutionary development of the application. The CM process also specifies procedures to report and resolve deficiencies discovered after versions are fielded.

Configuration Management (CM) is a discipline that uses both technical and administrative procedures to organize and control the development, implementation, and maintenance of software systems. The IRSS development team implements configuration management through a three part process that includes the following tasks:

- Configuration Identification
- Configuration Control
- Configuration Status Accounting

It is important to recognize that these three tasks are often implemented simultaneously and are continuous and iterative in nature. The configuration management process began with the release of IRSS Version 1 and continues throughout the operational life of the system.

Configuration Identification. The first task in the configuration management process is the identification of all configuration items that collectively comprise the system. Each unit of hardware, software, and documentation used in the design, development and operation of a software system is identified as a configuration item. All design documents, computer programs, and hardware items are identified individually to support effective change tracking.

Configuration Control. Once all configuration items have been identified, the second task in the configuration management process is to control any and all changes to these configuration items. Uncontrolled changes to any of these components may have serious repercussions on the development, operation, and/or performance of the system. Configuration management uses a variety of methodologies to control changes throughout the system's development life-cycle.

Configuration Status Accounting. The third task in the configuration management process is system status accounting. This task includes measures for reporting the status of the system, methods for verifying the completeness and comprehensiveness of the system, and procedures for ensuring the enforcement of CM policies and procedures. This task is accomplished through a series of regularly scheduled audits, reviews and status accounting reports.

2.2.1 Management

The Configuration Control Board (CCB) is the managing body for all proposed changes

to the IRSS project. The CCB for IRSS consists of representatives from the field sites and is chaired by a government representative from AF/XOR. IRSS' CCB team is the body for approving deviations and modifications to the original system requirements allocated to the product and identifying new requirements and functionality.

A designated representative of the Booz·Allen Team (the Configuration Manager) reports on the status of the Configuration Management efforts to the IRSS CCB. The current members of the CCB for IRSS are listed in Table 1.

Table 1 Configuration Control Board.

Chairperson	Maj Ken Moen AF/XORD
Board Members	Capt Brian Duffek SWC
	Lt Col Jeff Hewlett ACC
	Ms Barbara Mitchell AETC
	Capt Jay Mundy AFSOC
	Ms Janet Peasant AFRL
	MSgt Terry Robinette AFOTEC
	Mr. Jim Roe AFMC
	Maj Paul Summers AFSPC
	Capt Darryl Taylor AMC
	Mr. Bob Uhl AIA

2.2.2 Air Force CCB Roles and Responsibilities

The IRSS Air Force project manager approves the configuration baseline and is responsible for the final decision on any proposed changes to those baselines. The IRSS project manager must select proposed changes that warrant further study and eliminate those which do not fit in the scope of IRSS. The selected changes are submitted to the Booz·Allen CM for the development of an Engineering Change Proposal (ECP) and ECP Analysis, which is presented to the entire CCB for review. The analysis explains the items affected by the change and the projected level of effort for completion. It may also provide alternatives for the change. Changes are presented and discussed at CCB meetings. ECPs gaining a consensus vote (therefore approved) are submitted to the Booz·Allen CM for the development of an Engineering Change Notice (ECN). An ECN is the documentation required for the programmer to construct the change to IRSS. Finally, the CCB must select the implementation alternative for each ECP and prioritize all approved ECNs based primarily on cost, schedule and value added.

2.2.3 Booz·Allen CM Roles and Responsibilities

The Booz·Allen CM provides a media for IRSS users to report IRSS deficiencies and to propose new functionality to be implemented into future versions of the software. All received reports will be tracked from their receipt to their resolution through the IRSS ECP Tracking System (ETS). IRSS users input reports through User Reports located on the IRSS Internet web

site. The User Reports are then downloaded to the ETS. The ETS is a Booz·Allen developed application residing on the LAN and available to all members of the IRSS Team. The system is designed to simplify data entry and reporting of system anomalies and new functionality. The IRSS deficiencies will be corrected within IRSS and incorporated in future releases. Proposed new IRSS functionality will be forwarded to the IRSS Air Force project manager for selection of warranted updates. For the selected updates, the Booz·Allen CM will present the updates to the CCB for Review and a Vote. For approved ECPs, the Booz·Allen CM will develop an ECN which defines the agreed upon changes for IRSS implementation. The CM will issue each ECN to an IRSS programmer for implementation. The Booz·Allen CM will also provide status accounting reports that provide the IRSS community updates of all change requests. These updates include the status of outstanding ECNs and the composition and timing of upcoming version revisions.

2.3 Configuration Identification

This section describes the configuration identification process that establishes software baselines. Configuration Items are placed under configuration control when baselines are established. For IRSS, configuration identification includes identifying the documents that define the system during the development cycle, and providing the basis for production, testing, delivery, operation, and maintenance during the total system life-cycle. This section also describes the procedure for numbering and marking all configuration items.

2.4 Configuration Baselines

One of the key functions of Configuration Control involves defining the criteria for bringing an object under formal configuration control. An object may be a Software Product that is actually delivered to the client such as a Power Builder Application Module or a Requirements Document. An object can also be a Software Work Product that may or may not be delivered to the client such as a debugging or data migration tool. Table 2 depicts representative items brought under configuration management since Version 3.

Table 2 New items brought under Configuration Management control

Item	Description
Source Code Modules	Modules are brought under configuration control when the developers turn it over to the Software Quality Assurance team to begin the process of integration testing.
Test Plan	This document is brought under configuration control when the Software Quality Assurance Team circulates the test plan for review by the Team and obtains approval.
User and System Manuals	These are brought under configuration control when their authors circulate a draft for review and changes requested by Quality Assurance and Client

Item	Description
	Representatives are incorporated.
Change and Problem Fix Requests	Requests are brought under configuration control after they are approved by an authorized project staff (Team Leader or client representative) and sent to a developer. Booz·Allen will maintain a log of requests and track their progress.

Booz·Allen places source code modules under the control of PowerBuilder's version control tool after formal integration testing by the software quality assurance team and preliminary Beta Testing by the Program Manager and IPT representatives. The benefits of using a version control tool include: preventing two developers from updating the same version of module, simultaneously overwriting the changes made by the other, and ensuring that changes or bug fixes are made on the right version.

Several baselines have been identified for IRSS. The baselines are as follows:

- Functional Baseline
- Allocated Baseline
- Product Baseline

The baselines and products associated with each are described below.

- **MAJCOM Functional Requirements.** The MAJCOM Functional Requirements document defines the functionality that IRSS must contain to satisfy the participating field sites. The document includes all threshold and objectives for each IRSS version. The document defines the baseline requirements that must be implemented in IRSS.

- **Software Development Plan (SDP).** The SDP defines the process used to guide the development team in successfully completing a software project. It includes guidelines for Requirements Management, Software Project Planning, Software Project Tracking, Subcontractor Management, Software Quality Assurance, and Configuration Management.

- **As-Is Model Definition.** The As-Is Model Definition describes the functional requirements defined by the MAJCOM Functional Requirements. Booz·Allen developed the document to ensure that expectations for IRSS' capabilities are identical for all parties involved.

- **To-Be Architectural Assessment Design.** The To-Be Architectural Assessment Design provides the mechanism for describing how IRSS will implement user and system requirements from a design and architecture perspective.

- **Database Architecture.** The IRSS database architecture is defined by an Entity Relationship Diagram (data model) developed in the Erwin CASE tool. The data model

defines the relationships between entities captured in IRSS and defines how the data is stored in the database.

- **Data Element Dictionary (DED).** The IRSS Data Element Dictionary provides a listing of all discrete pieces of information captured in IRSS. The DED also provides a definition of those data elements and a list of example values for the element.

- **Test Plan.** The Test Plan provides the quality assurance team a methodical procedure for unit and integration testing for IRSS. The plan details how each IRSS component is tested and provides guidelines for testing future changes of IRSS.

- **Database Server.** The IRSS database servers, located at every participating site, provide the media for storing the database information and provides access for users to the database. The database server includes all the programs (i.e. Oracle, and Windows NT) and code needed for the server side of the IRSS client/server application.

- **User Interface Application Code.** The User Interface Application Code provides the look and feel for the IRSS system. It provides the user an easy way to access the database server for data input and retrieval. The User Interface Application Code resides on each client.

- **User/System Manuals.** The user/system manuals provide new and experienced users a source of reference for utilizing the capabilities of IRSS. The manuals provide detailed descriptions of all the functionality contained within IRSS.

2.4.1 Functional Baseline

The configuration Baselines are the agreed upon functionality documented and approved by Booz·Allen and the IRSS IPT for each IRSS version. The initial Baseline includes the MAJCOM Functional Requirements List, As-Is Model Definition, Software Development Plan (SDP), and statement of work describing versions 4 and 5. These documents provide the guidelines for the overall development of IRSS and supply in detail the functional requirements that constitute IRSS. The MAJCOM Functional Requirements and the As-Is document have come under configuration control after their presentation at the IPT meetings and delivery to Air Force Research Laboratory. The SDP, which is an internal document, has come under configuration control since the draft review by the Booz·Allen project manager.

2.4.2 Allocated Baseline

The Allocated Baseline is the second baseline established during systems development and documents the design the development team follows to build each IRSS version. The initial Allocated Baseline was established with the delivery of the To-Be Architectural Assessment and To-Be Addendum in February 1997. The database design provides a description of the storage structures used to contain the system data, the captured data elements, and the relationships between each data element. The Test Plan provides the guidelines for unit and integration testing for quality assurance.

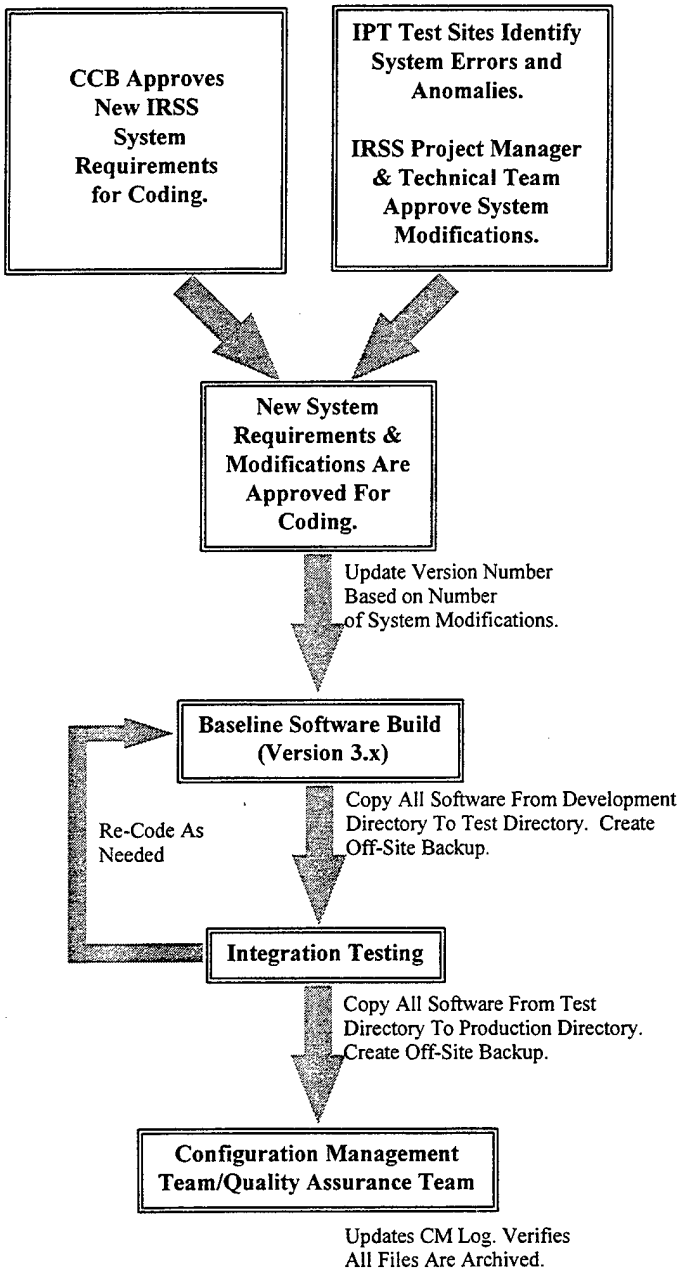
2.4.3 Product Baseline

The product baseline is the final baseline established during the system development life cycle and is established with the delivery of each IRSS Version. The current product baseline includes the software, hardware and user manual delivered with IRSS Version 3. The release of IRSS Versions 4 and 5 will follow under similar configuration management baseline control.

2.5 Configuration Identification Methodologies

The IRSS Development team uses a systematic approach for releasing new versions of the IRSS software. Booz·Allen places each release of the software under formal configuration management policies and procedures and tracks to completion. The software release methodology is described in Figure 1.

Figure 1 IRSS Software Release Process



New software releases may include new system requirements/functionality enhancements or modifications to existing code that does not operate properly. The IRSS IPT test site user community identifies new system requirements which are approved for coding through the Configuration Control Board (CCB) and its internal CCB process. The development team or the IRSS test sites typically identify the modifications to existing code as they perform system

functionality and usability testing. After the code modifications are approved, the IRSS version number is incremented to reflect the number of modifications from the previous IRSS baseline software product. Booz·Allen assigns the IRSS version number as follows:

- Version 1.0 - Contractual Deliverable
- Version 2.0 - Contractual Deliverable
- Version 3.0 - Contractual Deliverable
- Version 3.x - Where x is .1 for 10 or more system changes/modifications -or- x is .01 for less than 10 system changes/modification
- Version 4.0 - Contractual Deliverable
- Version 5.0 - Contractual Deliverable

Booz·Allen will perform periodic software releases and upgrades on an as-needed basis. Typically, the development team will release a new version of the software if improperly operating source code is identified during the test period. Booz·Allen will coordinate new releases and upgrades of the software with the IRSS Program Manager.

After the IRSS development team codes the software and unit tests the application, the source code, database and all associated files are compiled as a new software baseline. The new baseline files are copied from the development directory to the test directory in order to begin integration testing. The coding and integration testing process is iterative until the test team determines that the software operates properly. After the test team approves the software for production, a complete copy of the source code, database and all the necessary support files is archived off-site. The configuration management team and quality assurance team verify that the software is properly archived.

Throughout the coding phase, the development team places source code modules under the authority of a version control tool. The benefits of using a development environment with the capability to control source code object versions are significant, particularly with an object oriented programming language.

2.6 Configuration Control

After identifying all configuration items, the IRSS configuration management process tracks and controls all changes to these items against established baselines. The IRSS configuration control process controls four primary items:

- System software
- Documentation
- Hardware
- Database.

The IRSS CCB should approve all changes to these items. The following sections discuss IRSS configuration control procedures and the configuration control process for IRSS software, documentation, and hardware.

2.6.1 Change Control Procedures

There are four primary steps in the IRSS change control process. These steps, depicted in Figure 2, include IRSS assessment, team analysis, CCB meetings, and change implementation. These steps are discussed in detail in the following sections.

2.6.1.1 Client Assessment

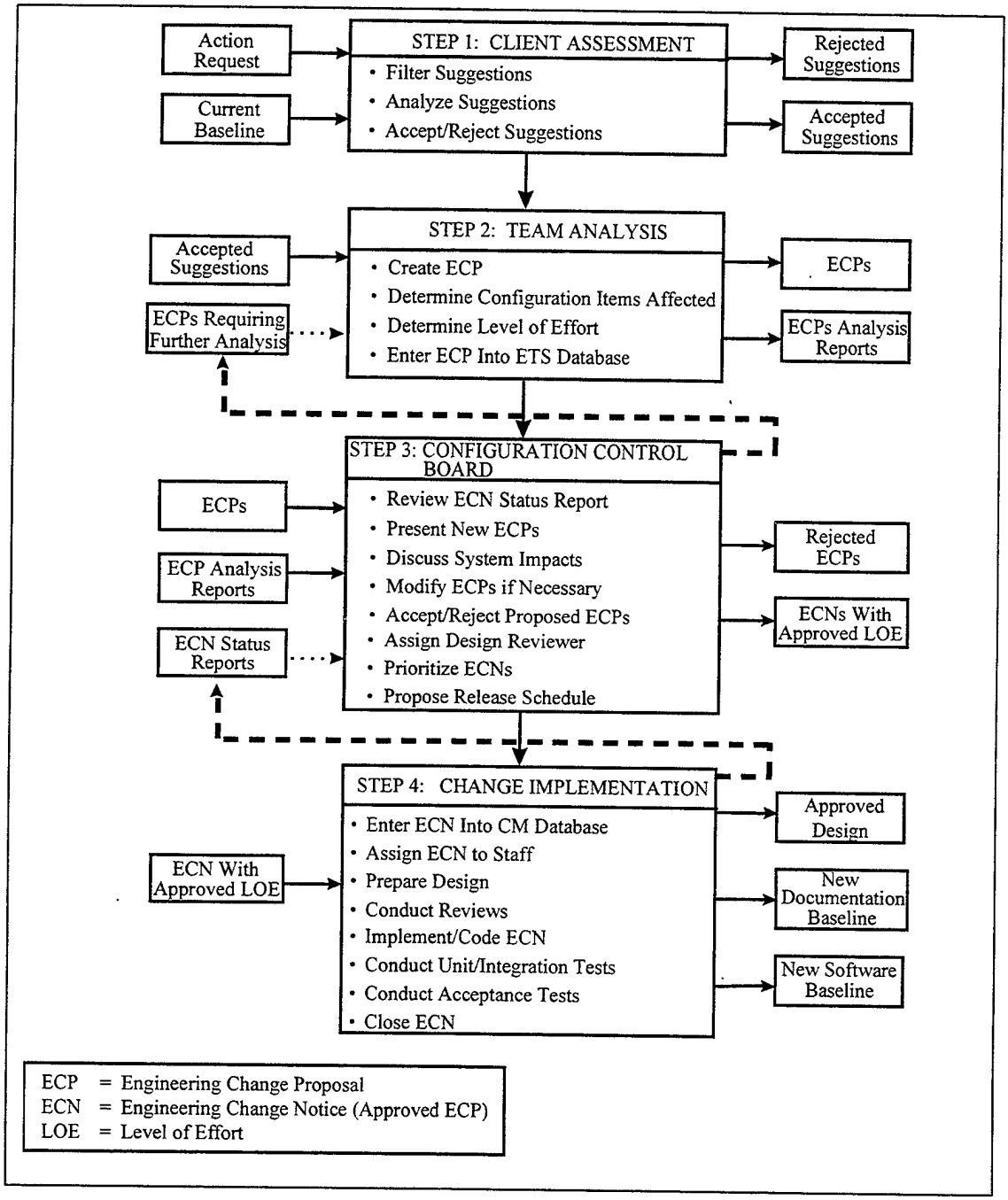
The IRSS change control process begins with the identification of a need for change by the IRSS development team, system users or other interested parties. There are two categories of changes:

- **Class I change** - Substantive changes to the system that introduce new functionality. These are classified as low, medium and high levels of effort.

- **Class II change** - Changes which result from testing or operational use which are required to make IRSS conform to the specifications.

The Booz·Allen Configuration Management Team has provided an IRSS User Report web site, which is accessible through any World Wide Web (WWW) Browser, to capture all IRSS change requests. The IRSS user report entry form provides an easy user interface to enter all change requests or user reports. When the user has completed entering the change request, the ***submit*** function on the Web Page will send the report directly to the Booz·Allen CM. The Booz·Allen configuration management team will import the submitted user reports into the ECP Tracking System (ETS) where the reports are separated. Class I requests and Class II requests. Class I requests are forwarded to the IRSS project manager for formal configuration management control beginning with the selection of warranted changes. The selected changes are signed and returned to the Booz·Allen CM for an in-depth analysis. The change requests that are not selected are terminated. Class II requests are implemented in IRSS by the Booz·Allen development team without the aid of the configuration control process. Figure 2 on the next page illustrates the change process.

Figure 2 IRSS Change Control Procedures.



2.6.2 Team Analysis

Class I changes which are returned to the contractor development team for analysis, become ECPs. All ECPs are entered into the IRSS ECP Tracking System (ETS). The ETS is an administrative system that allows the contractor configuration manager to track the progress of each ECP as it is implemented. The Development team assigns one or more team members to

investigate the ECP and methods of implementation. Air Staff members identify all configuration items affected, projects a level of effort to implement the alternative and suggests an implementation priority. The ECP is then presented at the next CCB. Helping the CCB make informed decisions is a cornerstone of the Booz·Allen CM process and will help ensure the successful and timely delivery of IRSS.

Booz·Allen has learned in the past that the secret to a successful CCB meeting is preparation, and our configuration manager will continue the tradition with IRSS by providing alternative methods for implementation. The effort required to identify alternative solutions pays off at the meeting because it enables the CCB to discuss not only the proposed change, but also some of the details associated with implementation e.g., level of effort, design considerations, etc.

2.6.3 Configuration Control Board Meetings

The CCB is an advisory board responsible for coordinating and controlling all changes to system baselines. The CCB is chaired by the IRSS IPT chairman and has a fixed number of voting representatives from XOR, Air Force Research Laboratory, and field sites which have a vested interest in IRSS. These representatives assist the chair in evaluating the impact and desirability of proposed changes. All meeting decisions are documented in minutes by a recording secretary.

After the commencement of the CCB meeting, the Booz·Allen development team will present the Engineering Change Notice (ECN) status report. This report lists the status of all ECNs, their implementation priority and the release version for which the ECN is scheduled for completion. The development team will also present new ECPs developed since the last CCB meeting. Each ECP is presented to the board for consideration. The board then discusses system impacts. If the CCB does not return the ECP to the development team for further analysis, Booz·Allen will approve, reject or modify the ECP. If the ECP is rejected no follow-up action occurs other than the update of the ETS to document the rejection. Rejected ECPs are not deleted from the ETS database. If the ECP is approved, by a CCB consensus vote and subsequent funding, an implementation alternative is selected, it officially becomes an ECN.

2.6.4 Change Implementation

After the CCB meeting, the development team updates the ETS database to reflect the decisions made during the meeting. Rejected ECPs are flagged while approved ECPs are marked as ECNs. Booz·Allen will assign Team members to implement the ECN. These team members prepare a design for implementing the ECN based on the chosen implementation option. Team members then conduct a design walkthrough with an IRSS design reviewer. Once the design is approved, implementation of the ECN undergoes standard development life-cycle procedures including the following activities:

- Design Walkthrough
- Code Walkthrough
- Test Plan Walkthrough

- Coding
- Unit Testing
- Integration Testing
- Documentation updates

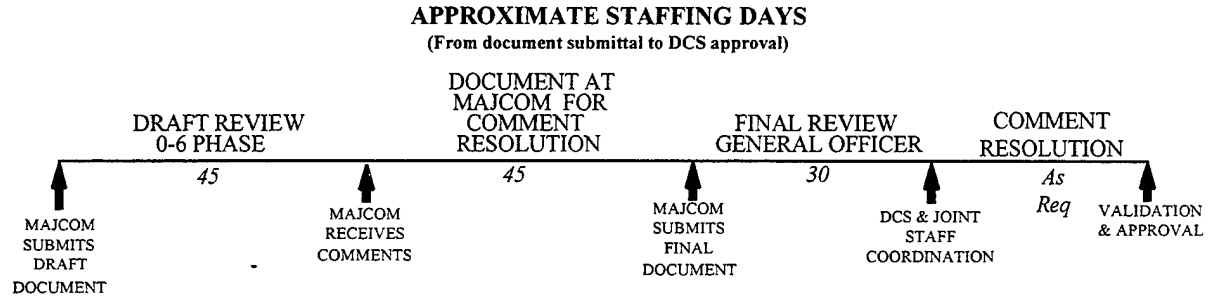
As the developers complete each ECN, it is included in the CCB status report as complete. The completed ECN is then delivered as part of the next software release.

3 IRSS Version 3.x Functionality

Section 3 discusses high level IRSS functionality to include its capabilities and the Air Force business processes supported by the application. IRSS is comprised of multiple modules that are linked together based on common information. The modules of IRSS include Projects, Documents, People and Executive Summary.

3.1 Air Force Requirements Process

AFI 10-601 specifies the iterative series of activities, to be performed by Air Force Action Officers (AOs) and many other DoD agencies, which formulate the acquisition process. The guidelines specify the steps required to prepare, validate, and approve MNS, Capstone Requirements Documents (CRD), and ORDs. The process, as depicted in the AFI 10-601, is presented below.



Source: Draft AFI 10-601, July 98

IRSS is designed and developed to support the acquisition process and the supporting activities performed by senior leadership and action officers. Working closely with members of the requirements community, Booz·Allen has captured the business processes to support their daily activities.

3.2 Using IRSS to Manage the Requirements Process

IRSS is designed to assist Air Force AOs in the management and preparation of information generated in the Requirements Definition Process. Specifically, IRSS modules manage the relationship between the people, projects, documents and tasks involved in the requirements definition process.

The following table lists the Air Force Requirements Process and provides a description of the IRSS process developed to support it:

AF Business Process	IRSS Capabilities
1. Identify a Deficiency	• Capture deficiency and descriptions • Capture Justifications (e.g., MAP, Congressional Language, National military Strategy, etc.)

AF Business Process	IRSS Capabilities
2. Address Milestone and Phases Criteria	• Capture summary information describing a project (i.e., Funding, participants, sponsors, ACAT Level, etc.). Provide link to justification • Provides link to supporting requirements documents and projects • Manage Project Schedule and tasks (e.g., Milestone Phases)
3. Draft Requirements Documents	• Capture document summary information (e.g., executive summary, participants, status, progress of documents) • Generate document templates • Provide ability to collaborate on document development • Perform full-text searches within local databases and across remote databases • Import documents and Graphics • Maintains RCM parameter linkages • Print documents and RCM matrix • Maintains document versions • Provide section traceability mechanisms (Including deficiencies and other document sections)
4. Perform Internal Review, Comment and Coordination	• Assign tasks to reviewers, with suspense dates and rationale • Provide real-time task notification • Tracks task status • Provide comments on-line
5. Comment Resolution and Document Approval	• Resolve comments on line • Generates comment resolution report • Captures approval authority signatures
6. Perform HQ USAF Staffing and Requirements Approval	• Coordinates document to external organizations for review and comment • Provides multi-level tasking ability • Tracks task status • Provides automated comment return

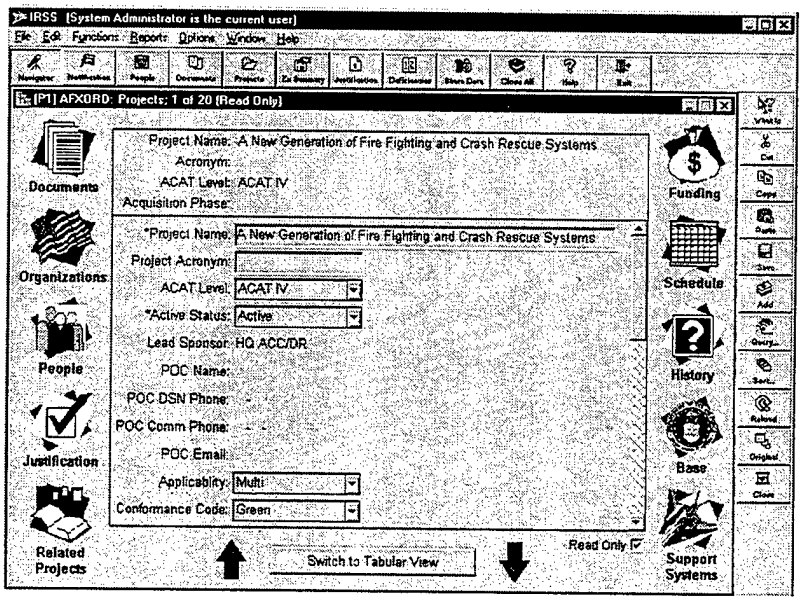
3.2.1 Project Module

The IRSS Project Module is designed to capture various types of data describing a project. The environment that IRSS is deployed within drives the type of data entered. For example, in the intelligence community, a project may represent a new threat. In the requirements community, a project represents the subject of a MNS or an ORD. In general, projects are initiated in response to a deficiency or need that has been identified and are usually

associated with the generation of supporting documents (e.g., MNS, ORD's, AoA's, etc.). The project module also captures or links to the following information:

Association	Description of Functionality
Documents	A link to all supporting or related documents within the IRSS document module.
Organizations	A link to all the related organizations within the IRSS People Module. Each organization may also be assigned a role description such as "Lead Organization" or "Test Organization".
People	A link to all people related to or supporting the project. Each link also provides the ability to supply a role description such as "MAJCOM POC" or "Responsible AO".
Justification	A link to Justification module which serves as the rationale for a projects inception.
Related Projects	A link to other related Projects within IRSS.
Funding	Captures decisions and funding allocations related to a project. Specifically, review board funding decisions and funding appropriations may be tracked in this section.
Schedule	Assigns an overall timeline for a projects progression through required phases. Each phase may be supported by multiple tasks and assigned to a resource(s) for completion.
History	Captures the significant changes that a project has gone through since inception.
Bases	A link to related or affected bases.
Support Systems	A link to related or supporting systems.

The project screen, depicted below in form view, displays the association icons described above.



Entering a Project and establishing its linkage to supporting rationale is usually the initial step performed by a user after the identification of a deficiency or need.

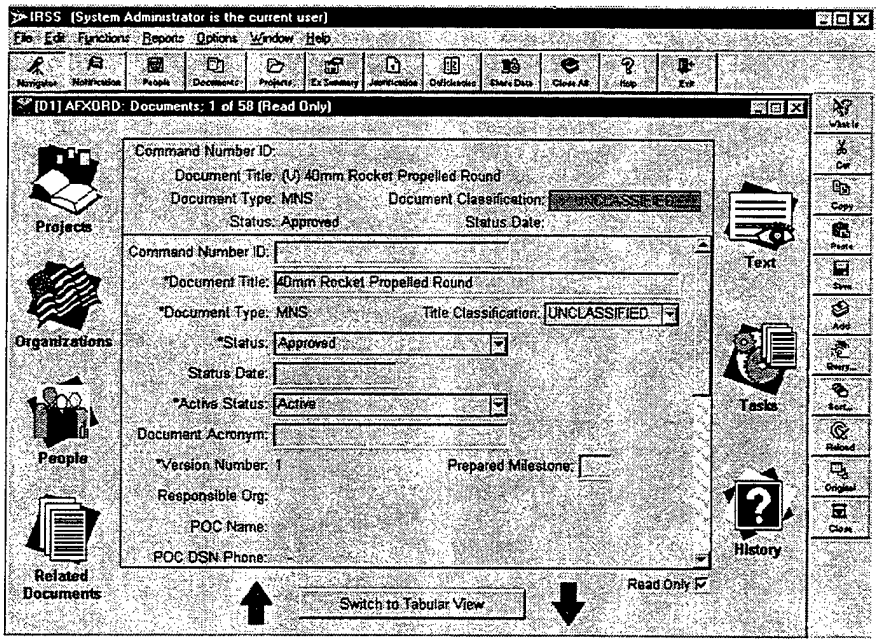
3.2.2 Document Module

The document module is designed to assist Air Force AO's in document management and generation. To this end, IRSS has the ability to capture and store any type of document. Like the project module, the documents module also captures and associates descriptive and supporting information. The document module captures or links to the following information:

Association	Description of Functionality
Projects	A link to related Projects within IRSS.
Organizations	A link to all the related Organizations within the IRSS People Module. Each organization may also be assigned a role description such as "Lead Organization" or "Test Organization" .
People	A link to all people related to or supporting the project. Each link also provides the ability to supply a role description such as "MAJCOM POC" or "Responsible AO".
Related Documents	A link to all supporting or related documents within IRSS.
Text	Opens the tree view of the document. Each section of the tree is generated according to the parameters assigned in the system administration module. Once generated, each section can be

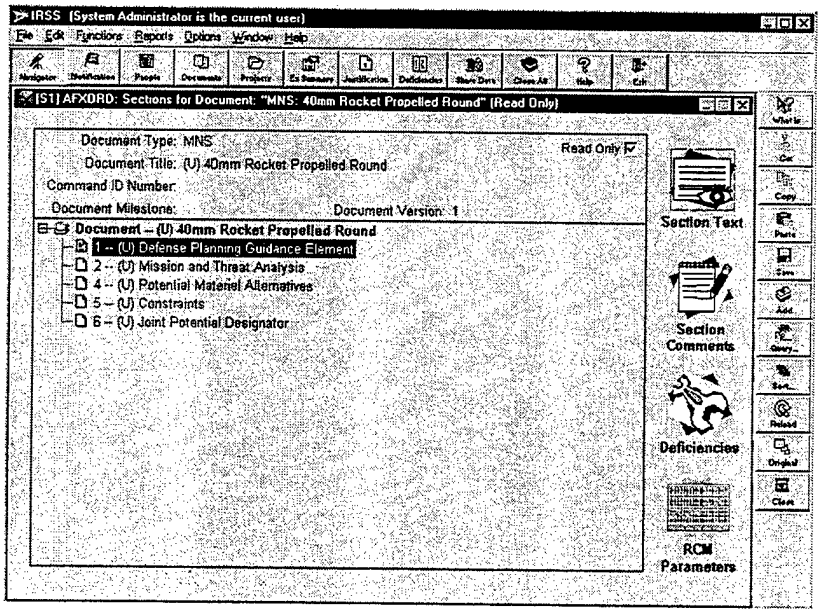
Association	Description of Functionality
	renamed, renumbered or additional section added. Each section contains the document contents or text.
Tasks	Opens the tree view of tasks assigned for the document. New tasks may be added and assigned to a resource(s) for completion. A resource can be a person, an organization or an IPT. This module includes suspense dates, planned start and planned completion dates among many others. Document tasks permit coordination of read-only copies of documents to users who reside on remote IRSS databases.
History	Captures the significant changes that a document has gone through since inception.

The document screen, depicted below in form view, displays the association icons described above.

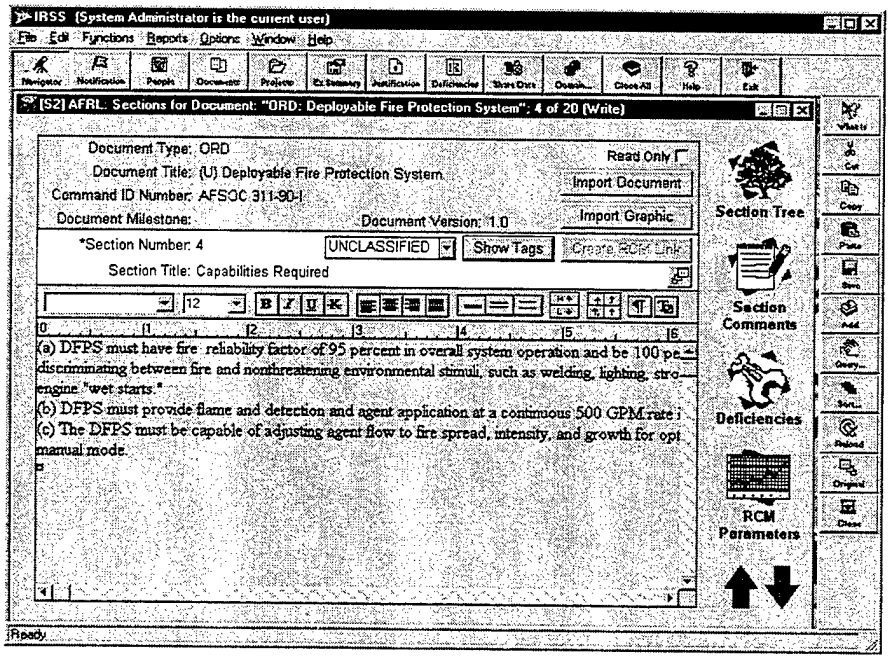


3.2.2.1 Using Document Text

The document tree is generated according to the document type established on the summary screen. Once generated, the tree structure can be changed according to the specific document requirements. The AO may add, re-number, and re-name document sections. The text contents of a document are saved in IRSS as separate sections. Each section corresponds directly to a typical document paragraph. In the tree view of a document, each section is displayed as an icon as shown in the following screen shot.

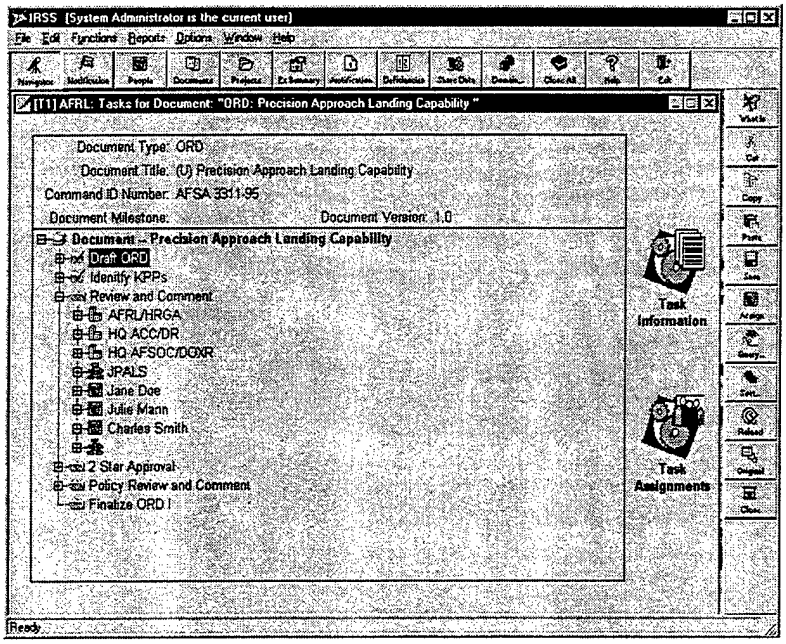


Once inside a section, IRSS provides word processing capabilities such as font choice, font size, bold, text color and many others. In addition, IRSS has a document and graphic import capability.

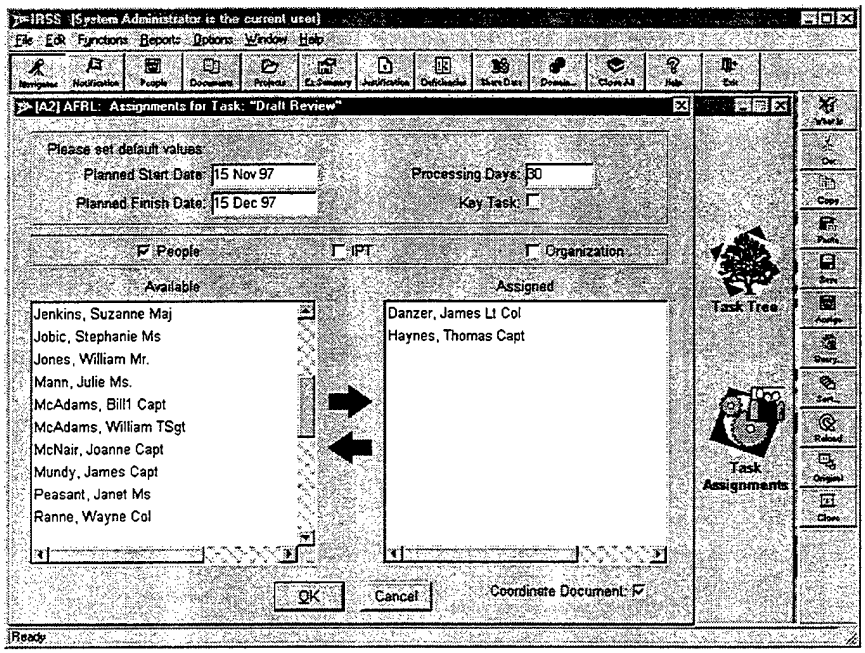


After a document has been drafted, it is sent for internal organizations to review and comment. A primary strength of IRSS is its ability to allow users to develop schedules and task resources to accomplish milestones within and across organizations. The task window, shown below, lists the task name and resources assigned in the tree-view format. IRSS models the Air Force coordination process and multiple layers of tasks can be assigned and one or more

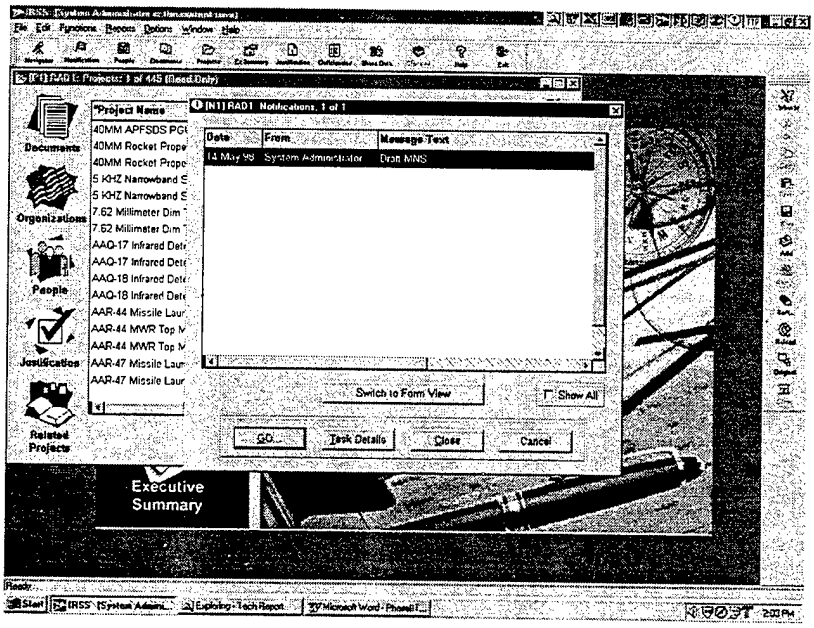
resources can be assigned to work on a task. The screen shot below displays a task list created for an ORD and the resources assigned for its review and comment.



The available resources list comes directly from the people module and includes people organizations, and IPT's. If the resource assigned does not reside on the local server, the "Coordinate Document" button becomes active. If checked, a read-only copy of the document is sent to the assigned resources' server. The server to server copy of the document enables AO's, from any location, to review and comment on the document on-line in real-time.

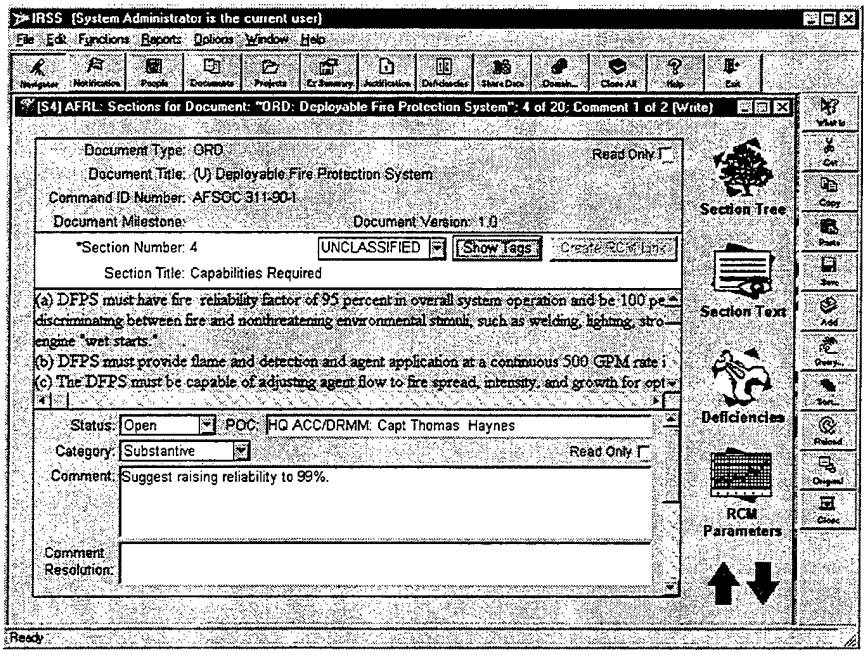


Once a task has been assigned and a resource allocated to work on it, a notification will be generated to the user. In the case of organizations and IPT's, the assigned POC's from their data card will receive notifications. The following notification window will appear when the resource signs onto IRSS or clicks the Notification button on the horizontal button bar.



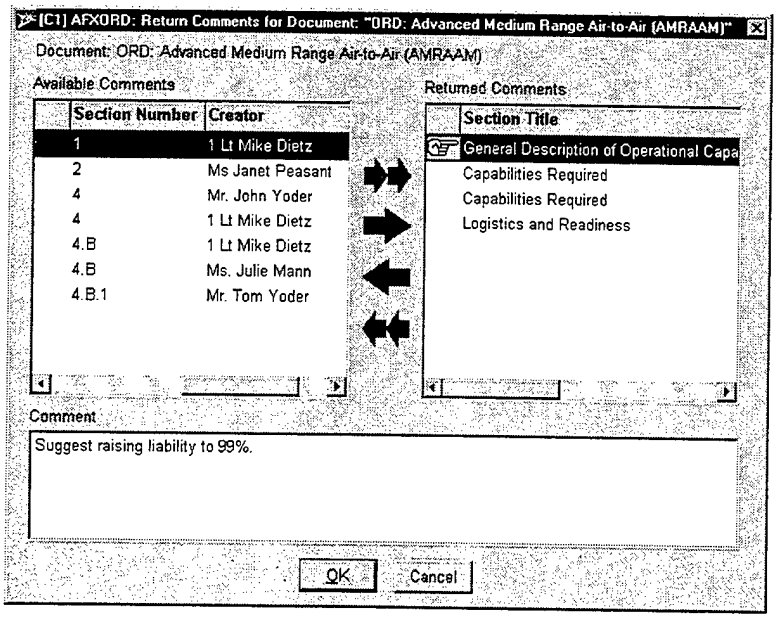
3.2.2.2 Using Comment and Comment Return Feature

During the Coordination phase of the acquisition process many organizations, and ultimately people, are requested to provide input on the same document and document section. The section comments screen, depicted below, provides the ability to view section text and provide comments in the same window.



Document tasks, that are coordinated to remote servers, will receive comments from one or more people. When all of the comments have been made, the remote POC has the ability to choose the comments which represent the organizations' position and return them to the original tasker.

The Document Comment Return Window, depicted below, is used to simplify the comment return process. This window contains three primary components. The Available Comments window located in the upper left of the frame window displays all the document titles, document sections and comment creators. If the user clicks on a particular row contained in this window, the Comment Window at the bottom of the frame window will be updated with the selected comment for that particular document section comment. It is possible that there will be more than one document title, section number and creator. This situation will occur if the same user creates more than one comment for the same document section or is supplied with security access to other comments.



After reviewing the comments, the AO can press the Move All button to move all the comments from the left window to the right window or they can select individual comments or multiple comments to be sent.

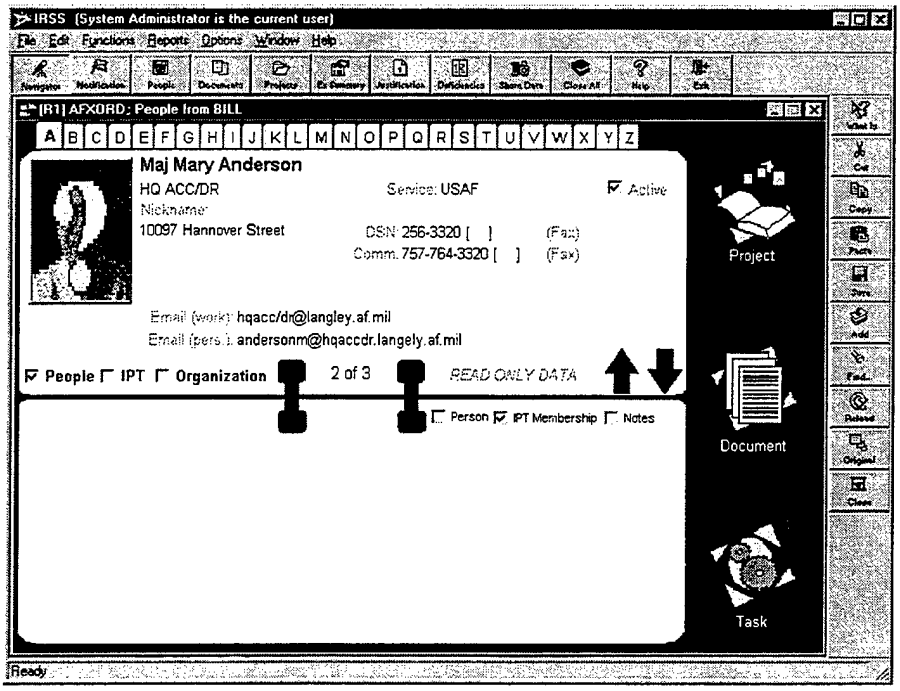
The selected comments are returned to the documents originating server, and will result in a "Returned Comments" notification to the original tasker.

3.2.3 People Module

The People module contains information about People, Organizations, and IPT's. This module serves as a single point of reference to contact or identify counterpart organizations throughout the IRSS community. This module also provides valuable workload information pertaining to each entry. The following table describes the workload associations that are available in the People module:

Association	Description of Functionality
Projects	A link to all related Projects within IRSS.
Documents	A link to all the related Documents within IRSS.
Tasks	A link to all related Tasks within IRSS.

The People screen below displays the association icons described above.



The People Module works like a cardfile. The check boxes located on the top half of the card switch the focus of the data being displayed. Each card includes the relevant data such as, e-mail address, etc. The People Module functionality is described below:

- An alphabet button bar is provided at the top of the screen. This bar allows the user to quickly access another section of the module by clicking on another letter.
- The card is split into two areas, the top half and the bottom half. The top half displays the main information about the selection. The bottom half displays information corresponding to the chosen view (i.e., person, IPT, or organization) and is used for data entry.
- The Project, Document and Task icons to the right of the window allow the user to view projects, documents and tasks that are linked to the person, organization or IPT currently displayed on the data card.
- The blue up and down arrow keys at the bottom right of the top half are used to go to the previous and next cards within the letter chosen.

3.2.3.1 Organization Cards

Combining a 'unit' and an 'office symbol' entry creates IRSS organization cards. An organization card can be made to represent a unit alone, such as USAF or can be defined as HQ ACC/DRMM. The 'unit' and 'office symbol' lists are defined in the system administration module and can be tailored to each IRSS domain. Organization cards require that a Point of Contact be specified. Specifying a POC is required so that at least one person receives tasking notifications. The POC field is a drop down list box that is generated from the people cards entered in IRSS.

3.2.3.2 IPT Cards

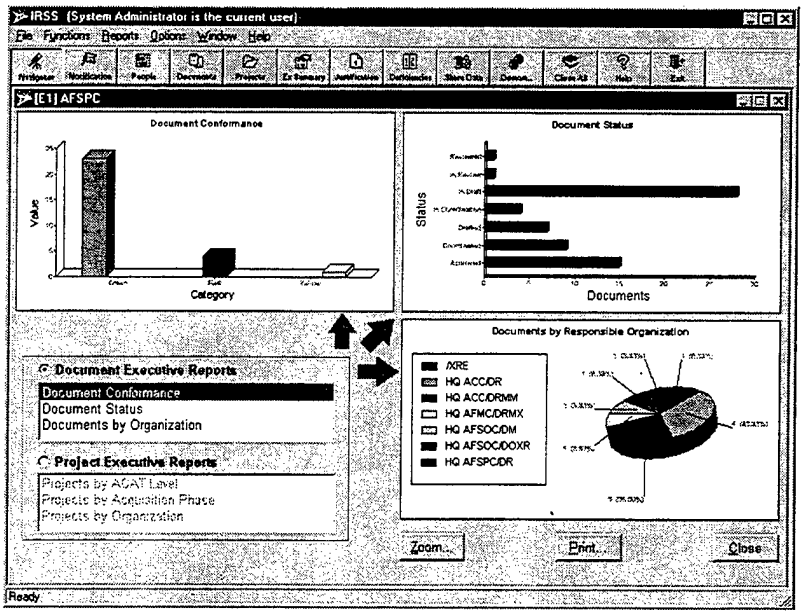
The IRSS IPT cards capture team descriptions, team types, and meeting dates as well as maintain the list of individuals serving on the team. IPT cards like Organization cards require a POC to be specified to receive tasking notifications.

3.2.3.3 People Cards

IRSS people cards require only a designator and last name to be specified as an entry. The unit/office symbol field on the data card is a drop down list that is generated by the organization cards contained within IRSS. Once a person has been assigned to an organization, the data describing the organization (e.g., address, phone numbers, e-mail, etc.) is copied to the people card. The people card can then be changed to reflect any additional or different data.

3.2.4 Executive Summary Module

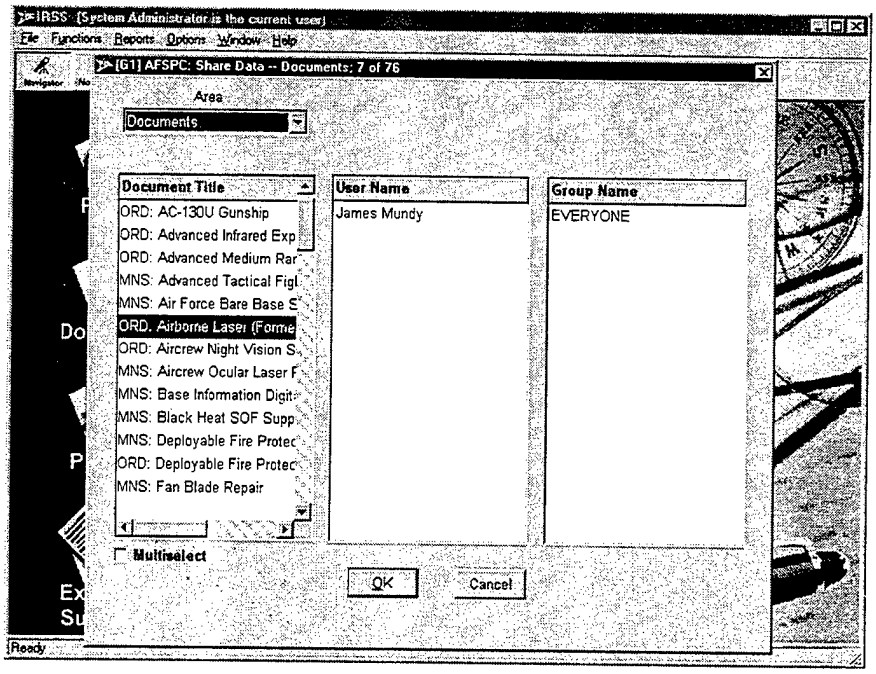
The Executive Summary module provides graphical reporting on documents and projects contained within IRSS. Only documents with 'active' status are displayed in the Executive Summary. Like other reports, the Executive Summary may be used to present data from multiple databases. The Executive Summary window is depicted below.



Each report can be modified to depict data in a different format or to be displayed with specific colors. Additionally, each graph includes a drill-down capability which assists in the rapid identification of potential problems and provides easy access to detailed project and document information contained within IRSS.

3.3 Share Data Module

All information created within IRSS is protected, by default, to the lowest level of security. For example, a document created within IRSS is available only to the creator until it is shared with other IRSS users. The share data module is the mechanism to permit individual users and groups of users to access documents; document sections, comments, and projects. The following screen depicts the share data module.



IRSS automatically assigns read-only access to users who have been tasked to work on a document or a project.

When a document is coordinated across servers, the person receiving the notification inherits read access to the entire document. The recipient can then task it out within their organization and grant other users access to view the data.

3.4 IRSS Version 4.0 - 5.0 Functionality

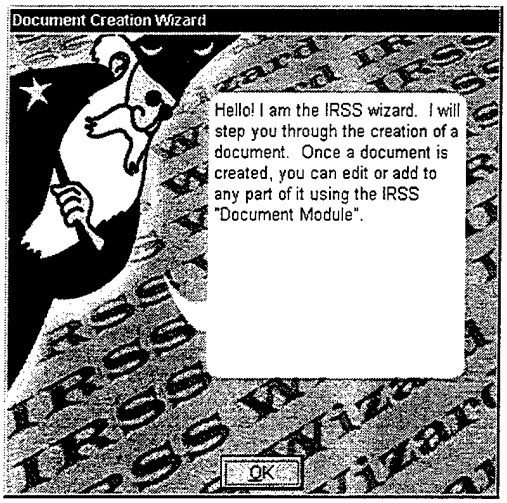
Future IRSS software releases will incorporate user requested enhancements and new functionality. The following list of enhancements will be released in the next two versions of IRSS:

- On-line IRSS tutorial
- Improved graphics and tabular data
- Improved data sharing capability
- Data delete capability
- DTIC coding
- N server architecture
- Classified/Unclassified integration.

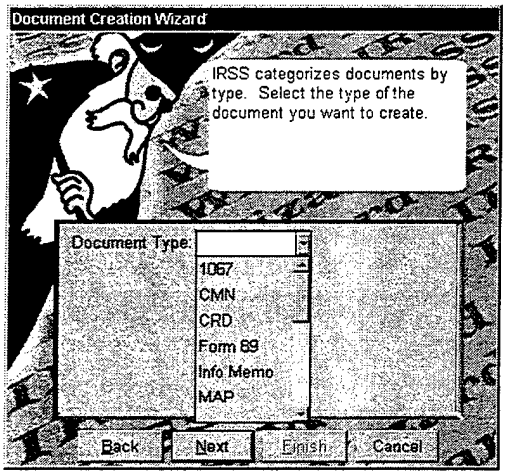
3.4.1 IRSS Tutorial - IRSS Wizards

IRSS versions 4 and 5 will include the first on-line, self-paced tutorial capability known as the IRSS Wizard. New wizards are being designed in accordance with increased functionality and will be incorporated in future releases. The wizards benefit novice and expert

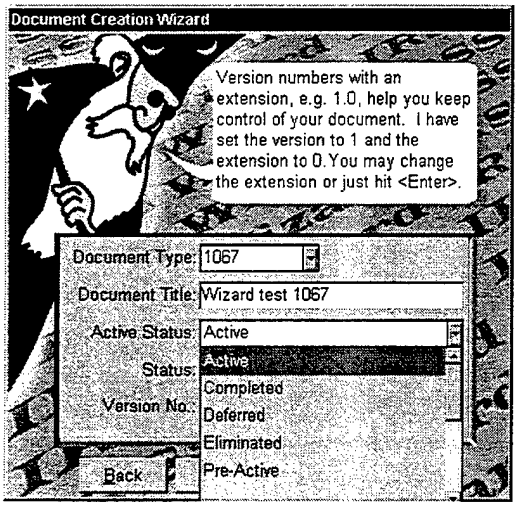
users by simplifying repetitive tasks and supporting the development of standardized data input. The initial wizard walks a user through the creation of a new document. The following screens depict the document creation wizard:



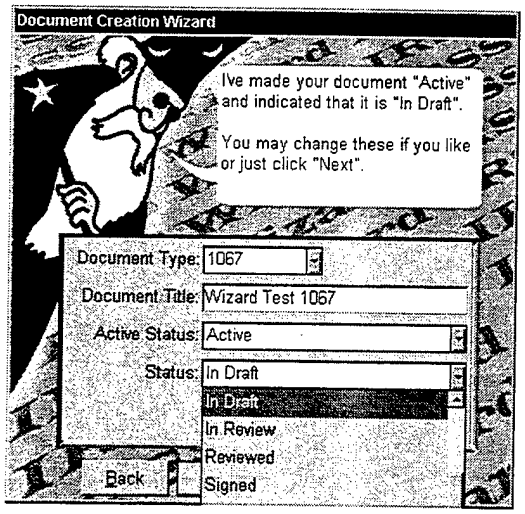
This opening screen appears when the user clicks on 'Create a Document.'



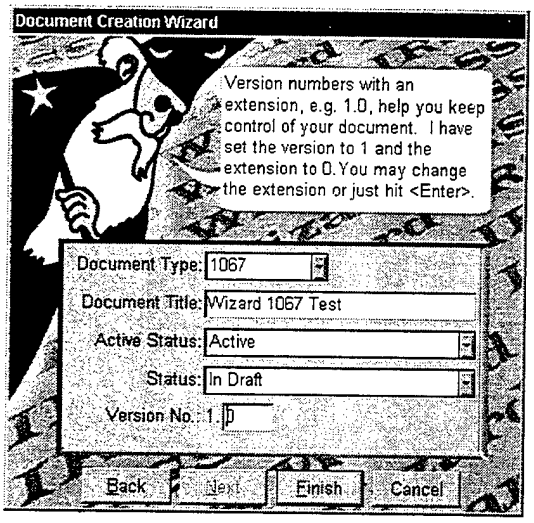
The next screen prompts the user for a document type using drop-down menus. The user can cancel the process at any time, or continue to the next screen.



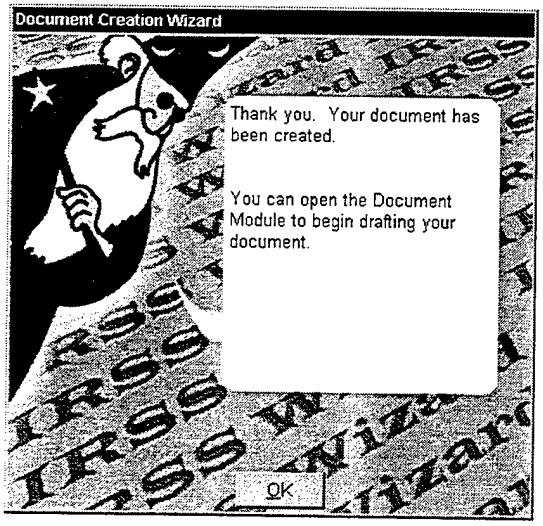
Continuing through the wizard, the Active Status Drop-down menu prompts the user to select the status of the document.



The next to last screen prompts the user to select the status of the document.



By assigning different versions to documents, you can track them separately.



The final screen tells the user where to go to view the document just created.

3.4.2 Tabular and Graphic Data

New to version 4 of IRSS is the addition of a text control feature from Griffin Technologies called TX Text Control. TX Text Control allows IRSS users to easily transfer information from IRSS text screens into common office automation applications. The screens have changed their look and feel to fulfill other requirements; however, TX Text Control performs all of its work behind the scenes. TX Text Control is a full-featured word processor custom control. It offers all the functions of a WYSIWYG word processor. This means the user has full control over the layout through a special view mode. Automatic pagination occurs during input and the pages are shown as they are displayed.

TX Text Control goes beyond the normal enhanced edit control. It includes sophisticated features such as:

- Substantial support for Microsoft Word-style Tables
- Import and Export of HTML files
- Enhanced RTF filter for better Word compatibility
- Streamlined file load/save interface
- OLE object embedding.

3.4.3 Improved Data Sharing

IRSS Version 5 will incorporate improvements to streamline the existing data sharing process. Current IRSS users must explicitly assign access privileges to data after it has been created. Several improvements can be realized by including user defined data sharing profiles which permit a one-time definition of common data sharing groups and users. The data sharing profiles will function similar to the retrieve profiles defined in section 3.4.4. It will enable automatic assignment of access rights to data as it is created. The existing Share Data module will continue to provide the mechanism to assign individual access to data after creation. We will continue to work with the IPT to identify other data sharing improvement opportunities.

3.4.4 Data Delete Capability

During the IRSS Data Delete Requirements User Session on 18 May 98, the IRSS IPT identified components of IRSS that should have delete or archive capability. Deleting data from IRSS will permanently remove the data from the database. Archiving data will transfer the data outside the active IRSS database environment, i.e., not retrievable from the IRSS interface.

The following candidates for deletion were identified and discussed at the session:

Error Correction Candidates	Function	Functional Roles	Problem Description	Issues/Comments
Documents	Archive	Sys Admin	Added a document in error. Multiple versions of same document.	Default display of all document versions confuses the user interface; difficult to identify unique/desired document. Archive similar to recycle bin concept. Data can not be retrieved from user interface
Document Sections	Delete	Sys Admin and Owner	Added a section in error	Incorporate an undo/delete option to removed undesired sections.
Notifications	Delete	Taskee	Notifications that are closed	Need to include the option of removing notifications that are closed if they are no longer desired.
People, Org's IPT's	Archive	Sys Admin and Owner	People, Org's, or IPT's entered in duplication	Incorporate an undo/delete option to removed undesired entries.
Document History	Potential future delete		Enter record in error	Currently can be edited to overwrite erroneous information.
Document Tasks	Presently Ok			Business Practice -Suggest Close task and note error.
Projects	Archive	Sys Admin	Same as document	
Project Phase	Archive	Sys Admin	Same as document	
Project Tasks	Archive	Sys Admin	Same as document	
Funding	Presently Ok			
Document Comments	Delete	Owner	Same as document sections	

User accessible deletes will be incorporated for the following items: notifications, document comments, document sections, and document and project history records.

Retrieve profiles will be used to satisfy the need to "delete" or "archive" records which may have been coordinated or synchronized across multiple databases. A field called "Activity Category" will be added to the document and project summary interface. The "Active Category" field will be a drop down list-box and will include the following options: In Process, Approved, or Inactive, with the exception of the People Module which will only allow Active or Inactive settings. The People module settings will be accomplished with a check box located on the cardfile card. When added to the system, documents and projects, default to In Process. Only users with supervisor or administrator access have the ability to change status's.

A recycle-bin feature will also be added to the system and will be called "delete". The feature will move the data into a recycle-bin table which can not be retrieved via the IRSS interface. All records moved into the recycle-bin table will need to be periodically deleted by a database administrator once they have been safely and without impact from the IRSS network. Only system administrators will have the ability to "delete" or "recycle" data.

User profiles will be added to the system to allow user defined settings of retrieve preferences. The user profile preferences will include the following options:

- Documents: In Process, Approved, Inactive
- Projects: In Process, Approved, Inactive
- People: Active, Inactive
- Executive Summary: In Process, Approved, Inactive.

The user profile will include a "Check with me Before Retrieve" option which, when activated, will prompt the user to define retrieve selections before responding with updated information.

3.4.5 Defense Technical Information Center (DTIC) Distribution Coding

DTIC coding will be incorporated into the IRSS tables requiring a level of distribution coding. As an element of the Defense Information Systems Agency (DISA), DTIC provides access to and facilitates the exchange of scientific and technical information thereby contributing to the management and conduct of Defense research, development, and acquisition efforts. In a nutshell, DTIC provides information--records of planned, ongoing, or completed Defense-related research--to U.S. Government agencies and their contractors. DTIC distribution coding is guided by DoD Directive 5230.25, "Withholding of Unclassified Technical Data from Public Disclosure" Criteria for Withholding. The Directive provides that data may be withheld from public disclosure when all of the following criteria are met. The technical data:

- Are in the possession of or under the control of the Department of Defense
- Have military or space application
- May not be exported lawfully without an approval, authorization or license under U.S. export control laws
- Disclose critical technology.

Information under the control of or in the possession of the Department of Defense means data created or received by elements of the Department and information developed and produced for the Department under contractual arrangements or other agreements. All documents produced in finality must contain a distribution code of one of the following categories:

- **Distribution Statement A:** Approved for public release; distribution is unlimited.
- **Distribution Statement B:** Distribution authorized to U.S. Government agencies only. Other requests for this document shall be referred to controlling DoD Office.
- **Distribution Statement C:** Distribution Authorized to U.S. Government Agencies and their contractors. Other requests for this document shall be referred to the controlling DoD office.
- **Distribution Statement D:** Distribution to DoD and DoD contractors only. Other requests shall be referred to controlling DoD office.

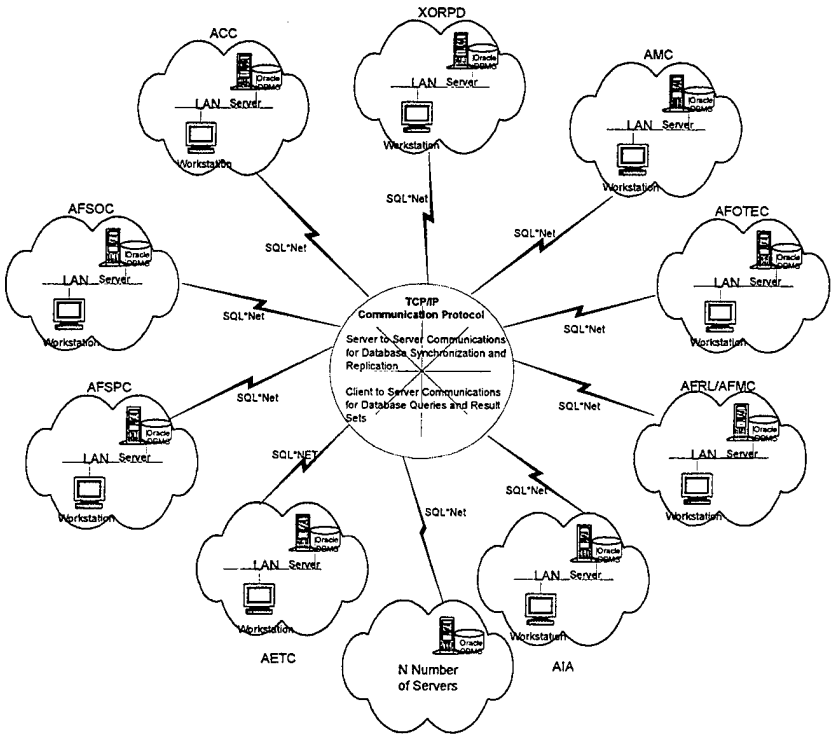
- **Distribution Statement E:** Distribution Authorized to DoD components only. Other requests for this document shall be referred to the controlling DoD office.
- **Distribution Statement F:** Further dissemination only as directed by controlling DoD office or higher authority.
- **Distribution Statement X:** Distribution authorized to U.S. Government agencies and private individuals or enterprises eligible to obtain export controlled technical data in accordance with DoD Directive 5230.25.

3.4.6 N-Server Architecture

As IRSS migrates from a prototype to an operational system, numerous system and maintenance related features will be added and enhanced over time. First, the Oracle stored procedures that control replication and coordination will be revamped to support N-server architecture that will ease future code maintenance as more sites are added. The significance of N-server is that the client and RDBMS software support adding servers without intensive maintenance on code. Currently, IRSS is a tightly coupled federation of clients and servers. Each address of each server is hard coded into various replication and coordination modules. Adding servers is a code maintenance intensive task.

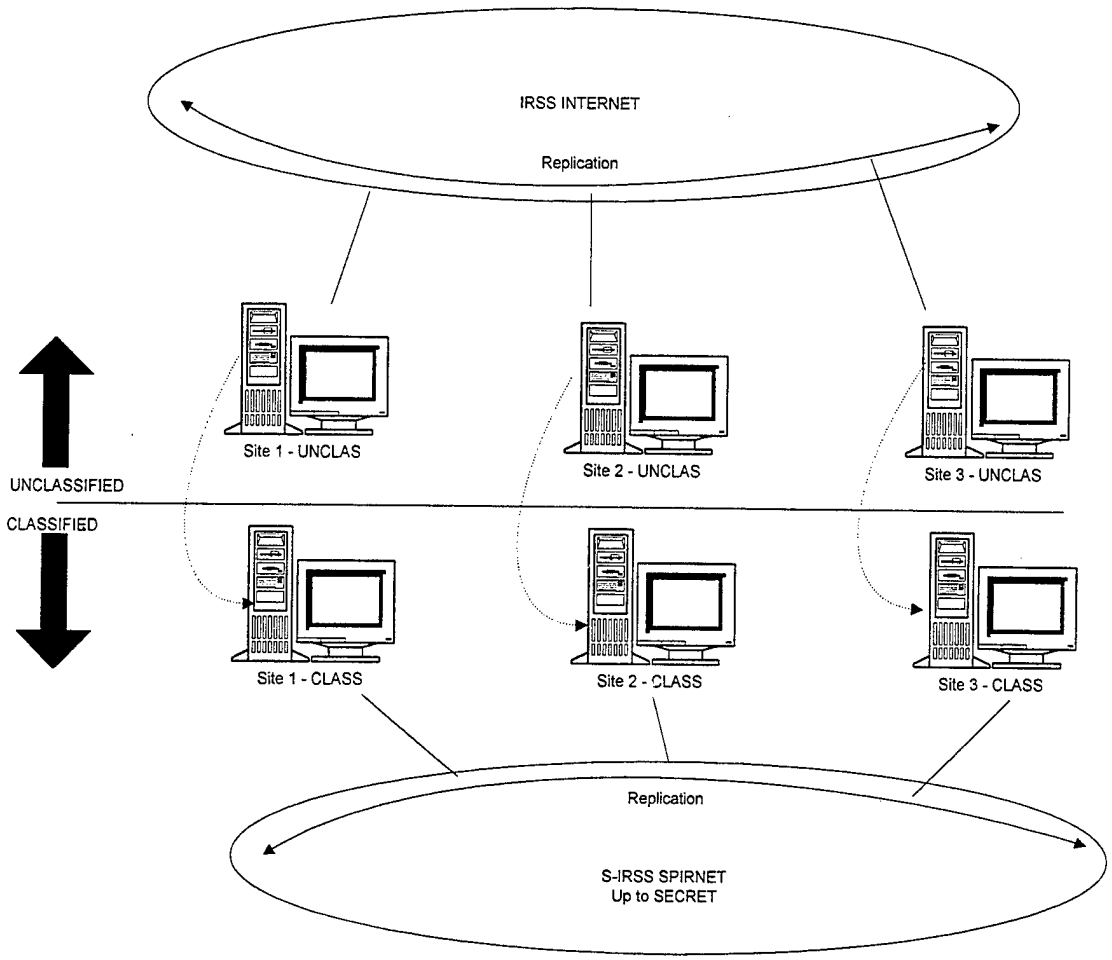
With N-server architecture, hard coding of server addresses will evolve to a dynamic environment using the ***tnsnames.ora*** file to store information about all servers. This means that as servers are added to the IRSS federation, their addresses are added to a Master ***tnsnames.ora*** file. All references to servers in the code will be by a variable that references the ***tnsnames.ora*** file for addressing and information about servers. These variables are different for coordination and replication so that both operations can operate simultaneously, referencing information about servers from the ***tnsnames.ora*** file and not confuse each other's directions for which tables to be replicated or coordinated across the federation. This enables the future growth of IRSS without changing the code.

The following diagram illustrates the IRSS federation and locations, and depicts the ability to scale the number of servers to meet the user's needs.



3.4.7 Classified/Unclassified Integration

IRSS Version 5 will allow for the interaction between classified and unclassified IRSS databases. Several candidate architectures and interaction mechanisms were evaluated by the IPT user community. The following diagram depicts the goal architecture as defined.



This architecture represents two parallel operating environments, one classified and one unclassified. The dotted lines represent the data transfer mechansim to be developed.

4 Technical Architecture

Section 4 discusses IRSS' technical architecture to include the development environment and system architecture. This section also details technical aspects of IRSS functionality and client and server operation.

4.1. Database Environment - Oracle

IRSS consists of a PowerBuilder 6.0 based client application and multiple databases implemented using Oracle Workgroup Server Version 7.3.2 and SQL*Net version 2.3.2. Each IRSS database is composed of groups of database objects such as tables, triggers, stored procedures, indexes, snapshots, and snapshot logs.

IRSS operates in a distributed environment in a 2-tier configuration. At a high level of design, the IRSS database business logic can be summarized into coordination and synchronization. In the IRSS context, synchronization is sometimes referred to as 'replication.'

IRSS logical database design is the formal data specification and is comprised of the following two documents:

- IRSS Entity-Relationship Diagram (logical data model)
- IRSS Data Element Dictionary

The Entity Relationship diagram presents a graphical representation of the entities and their relationships (logical model) for the IRSS application. The diagram was developed using the ERwin CASE tool. ERwin can also generate the SQL statements needed for defining the physical model on the Oracle database server. Specifically, the Entity Relationship diagram captures the following pieces of information:

- Entity Names and attributes (and their domains)
- The identifier for each IRSS entity (the subset of attributes that uniquely identify an instance of an entity—through the use of a primary key)
- The various relationships between the entities (cardinality).

In addition to the use of attributes that fully capture all its properties, every IRSS entity will contain special attributes that can create and maintain multiple versions of the entity as well the Date and User ID of the last update to that entity. The logical model is the language suited to specify the logical database structure for the Oracle RDBMS.

The IRSS data dictionary, which is generated using the IRSS Entity Relationship diagram, gives a narrative description of the various attributes associated with IRSS entities. Specifically the data dictionary is a repository of information about the entities and attributes of a database and contains the following pieces of information:

- Entity—the name of the entity, connected to the Entity Relationship diagram. This will

occur as a table in the Oracle database implementation of the design.

- Attribute—the name of the fields in the table. The IRSS-R Standards document describes data attribute naming standards. This standard uses primary data + <modifier>+ class word, e.g., EMPLOYEE_LAST_NAME.
- Description/Comment—a text description of the contents stored in the attribute and any supplemental comments. The supplemental comments will include the Edit/Integrity rules for attribute values.

4.2 Developer Environment – PowerBuilder 6.0

New to versions 4 and 5 of IRSS is an upgrade to PowerBuilder 6.0. PowerBuilder is a comprehensive development environment for building high-performance, client/server applications. It combines graphical painters with object-oriented programming language and provides enterprise connectivity, with native access to client/server databases. Other features include 16- and 32-bit capabilities, full object orientation and a shared application library and check-in/check-out capability.

4.3 Systems Requirements

The IRSS system architecture consists of a client/server technical implementation developed by assessing and prioritizing business and information requirements, evaluating technical alternatives, and assessing technical and cost factors.

The system architecture hardware specifications consist of the following components:

Client Workstation - Hardware and Software Components

Hardware	Specifications
Client (Workstation) Operating System	A 32-bit OS (Windows 95 or NT)
Client Connectivity to Oracle database	SQL*Net TCP/IP Client
Client Hardware—CPU	Pentium
Client Hardware—RAM	16 Mb minimum
Client Hardware—Disk Space	20 Mb of free disk space
Client Hardware—Monitor	SVGA, Resolution 800 x 600, 256 Colors

Database Server - Hardware and Software Components

Hardware	Specifications
Server DBMS	Oracle Workgroup Server running Oracle 7.3.4 (1 user license)
Server Operating Systems	Windows NT 4.0 or higher (10 user license)
Connectivity	TCP/IP
Network Card	3COM 3C90x PCI 100MB Network Card
Server Hardware—CPU	Intel PentiumPro 200MHz Processor
Server Hardware—RAM	Minimum Configuration 32 Mb required—has to be scaled up depending upon number of concurrent users
Server Hardware—Disk Space	2 GB IBM Wide SCSI hard drive, sufficient to support the database server kernel, on-line documentation, and the IRSS database
Server Hardware—CD ROM	ATAPI CD ROM Drive
Server Hardware—Tape Drive	Colorado T4000 Tape Drive
Server Hardware—Keyboard	101/102 Keyboard
Server Hardware—Mouse	Microsoft Mouse
Server Hardware—I/O Controller	Intel Chipset Wide SCSI Controller
Server Dimensions	17 x 10 x 18 (H x W x D in inches)
Server Monitor Dimensions	18 x 17 x 20 (H x W x D in inches)
Server Keyboard Dimensions	19 x 7 x 2 (L x W x H in inches)
Storage Environment	Proper temperature control, humidity control, and ventilation to support a server

The IRSS client/server architecture provides several performance benefits, including:

- *OOP/OOD Environment*—Powerbuilder provides full support for object oriented design and implementation
- *Extendibility*—Flexibility and ease of modifications allows for extendibility to other applications
- *Reliability*—Incremental modular development and compliance testing ensuring system reliability
- *Usability*—GUI design provides industry standard appearances and interfaces
- *Integrity*—Controlled access to system functions based on user privilege ensures integrity

- *Correctness*—System increments meet the requirements defined by each organization
- *Efficiency*—Rapid system response time and appropriate utilization of system resources makes IRSS efficient
- *Adaptability*—Flexibility to rapidly incorporate unanticipated and future requirements
- *Verifiability*—Compliance tests to determine how each system increment will handle specific situations allows for isolation and identification of errors, which leads to a system that is verifiable.

However, the implementation and management of a client/server architecture based system also presents a number of complex performance management issues and concerns. The number and variety of technologies that constitute IRSS make it difficult to gather, measure, and tune the system with one centrally located tool or utility. It requires a number of tools, utilities and intuitive methods to pinpoint problems or potential performance issues. It is the responsibility of the System Administrator and the Database Administrator (DBA) to tune and operate the system for efficient use. This usually translates into properly monitoring and tuning the four basic resource components of the system namely, server memory, disk I/O, CPU, and network traffic. The Oracle RDBMS and the Windows NT operating system provide a number of tools, utilities and scripts to monitor and tune system performance. Some of the more commonly used tools and utilities include:

- SQL*DBA monitor (ORACLE utility)
- SQL TRACE (ORACLE utility)
- TKPROF (ORACLE utility)
- UTLBSTAT.sql and UTLESTAT.sql (ORACLE scripts).

Tuning the system is an iterative process involving testing, monitoring performance, modifying system HW/SW parameters and re-testing. IRSS's performance and tuning goal is to achieve the following:

- Memory utilization close to 100 percent (thought it may have to be scaled up depending upon number of concurrent users)
- High performance I/O rates (achieved by evenly distributing the I/O across the various disks, and other techniques)
- Restrict paging and swapping to a minimum.

4.3.2 Server Enhancements

There are no major architectural changes to server requirements in the next release of IRSS other than those made to correct error conditions in the application that cause changes to be made to the server.

These relatively minor changes resulted in minor architectural adjustments that may be reviewed and updated under later versions. They are however, fixes that required both client user-interface changes and table and view modifications on the server.

4.3.3 Client Enhancements

A major change to the client side of IRSS is the upgrade of the development software to PowerBuilder 6.0. This upgrade allowed us to make substantial changes not only to the look and feel, but to the functionality of IRSS, i.e., Microsoft standard menus. For example, enhancements have made it easier to import tabular data supporting Rich-Text Format version 3.0. The Document Section has been revised to provide an expanded working area. The Ad Hoc Query table has been adjusted to allow for a better user-interface. The Comment and Resolution Report has been modified to ensure automatically growing fields don't overlap each other. The Personnel Roster is now able to display commercial telephone numbers, and the Task trees now display name and rank.

One of the biggest gains and more creative features is the development of IRSS Wizards to serve as an on-line tutorial. Wizards have been used extensively and very successfully, by Microsoft and other leading software vendors. We feel wizards enable all users from novice to expert to benefit by simplifying the tasks and creating standardized documents and reports. We have added our first wizard for Creating Documents and anticipate many more with increased functionality in each one, in the major version updates for Versions 4 and 5.

4.4 Coordination Enhancements

Coordination is achieved by copying local data to other remote database sites. In IRSS context, this process is called 'PUSH' process. When a user coordinates a task, data in the local database is copied to an associated temporary holding table. Next, the data in the holding table is propagated to tables at other IRSS database sites. The data is propagated to other IRSS database sites at a regular time interval, managed as a database queued job. In order to coordinate RAW data (e.g., images, non-text, binary, etc.) data a one time retrieve is required by the client.

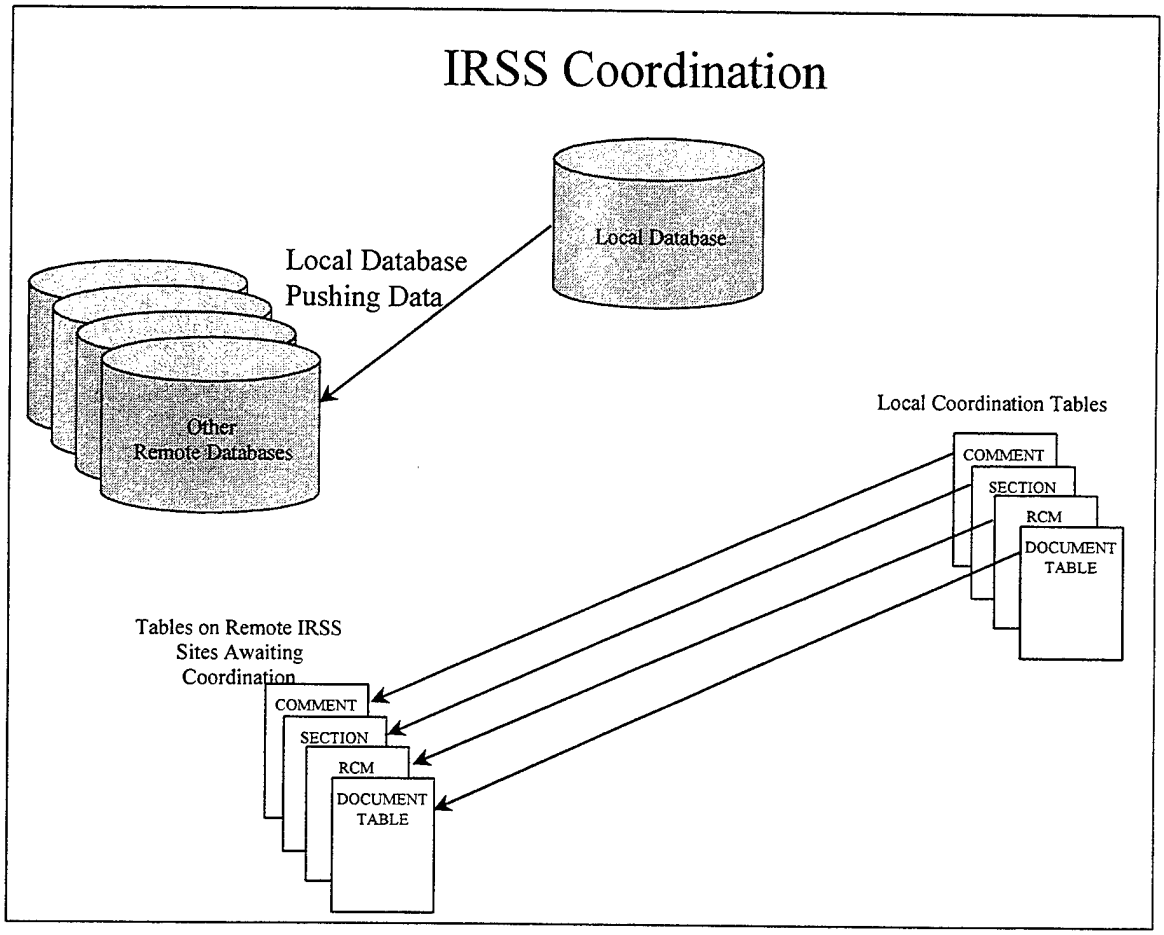
Currently, data in the tables COMMENT, DOCUMENT, RCM, and SECTION are copied onto holding tables COMMENT_COORDINATION, DOCUMENT_COORDINATION, RCM_COORDINATION, and SECTION_COORDINATION, respectively, for propagation to other IRSS sites. As the data is propagated successfully, the copied data in holding tables is deleted in preparation for the subsequent propagation. The PUSH process is performed by executing a series of custom Oracle stored procedures.

The following top-level procedures are used in this coordination process (Note that XXXXX represents the acronym of IRSS organizations, i.e. AFXORD):

- PUSH_XXXXX – The top level procedure that initiates data propagation.
- TNS_PING – This procedure tests for proper network connection between IRSS databases prior to data transmission.
- COPY_TO_DOCUMENT_XXXXX – This procedure pushes data in local DOCUMENT_COORDINATION table to remote DOCUMENT table.
- COPY_TO_SECTION_XXXXX – This procedure pushes data in local SECTION_COORDINATION table to remote SECTION table.

- COPY_TO_RCM_XXXXX – This procedure pushes data in local RCM_COORDINATION table to remote RCM table.
- COPY_TO_COMMENT_XXXXX – This procedure pushes data in local COMMENT_COORDINATION table to remote COMMENT table.

The following diagram depicts coordination in IRSS.



4.5 Replication

IRSS currently utilizes a basic replication technique called read-only snapshot. A read-only snapshot is a full copy of a table, or a subset of a table, that reflects a recent state of the master table. An IRSS snapshot is defined by a distributed query that references one master table. A database that contains a master table is referred to as the master database.

Each replica (or copy) of the master table is called a snapshot because the information captured at a moment in time can be periodically refreshed to reflect a more recent transaction-consistent state of the master table.

A simple snapshot is based on a single remote table and has none of the following: distinct or aggregate functions; GROUP BY or ORDER BY clauses; sub-queries; joins; or set

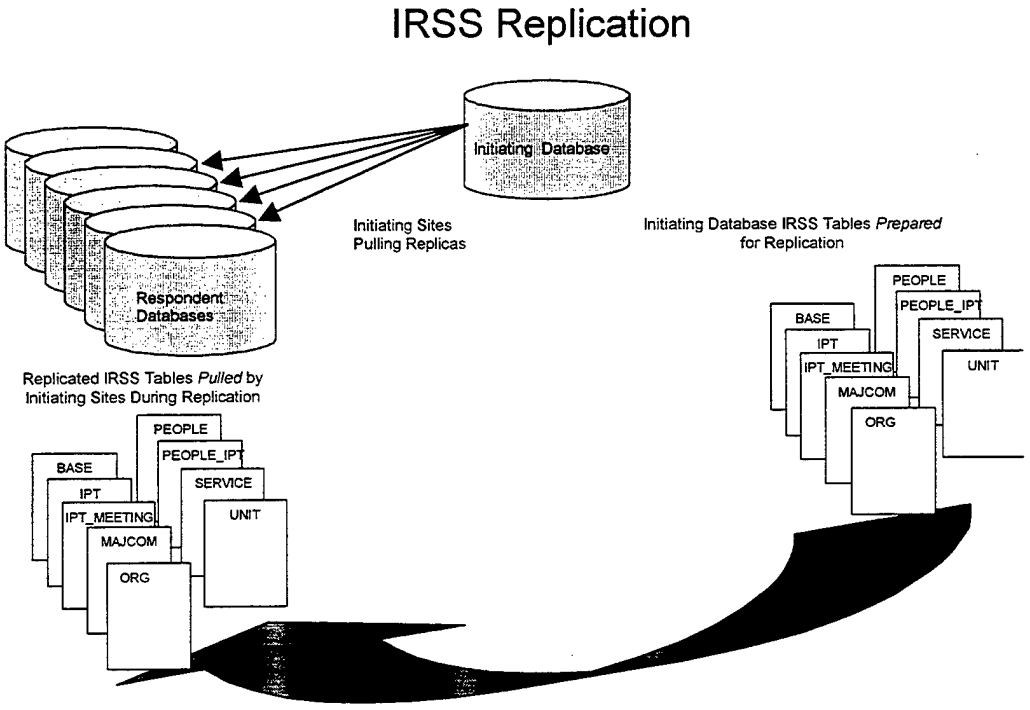
operations. If a snapshot's defining query contains any of these clauses or operations, it is a complex snapshot.

For simple snapshots, you can choose to create a snapshot log for the master table. This log is named MLOG$_master_table_name and the trigger used to update this log is named TLOG$_master_table_name. The information in this log allows you to perform a fast refresh of a simple snapshot.

With a fast refresh, only the changed rows of the snapshot, as indicated by the snapshot log, need to be updated. Each time that you make a change to the master table, Oracle tracks that change in the snapshot log, including the ROWID of the changed row. The generated index (I_SNAP$_) on the ROWID column of the base table allows these changes to be quickly applied to the snapshot. A complex snapshot, or a simple snapshot without a snapshot log, must be completely regenerated from the master table every time you refresh the snapshot. This is known as a complete refresh.

In IRSS, following database tables are replicated to one site from all other sites: BASE, IPT, IPT_MEETING, MAJCOM, NOTIFICATION, PEOPLE, PEOPLE_IPT, SERVICE, ORGANIZATION and UNIT. Once the table data are replicated, local table triggers are used to move data from snapshot tables to IRSS core tables. In the IRSS context, this process is called 'PULL' process. Currently, replicated data is refreshed in regular interval as a database queued job.

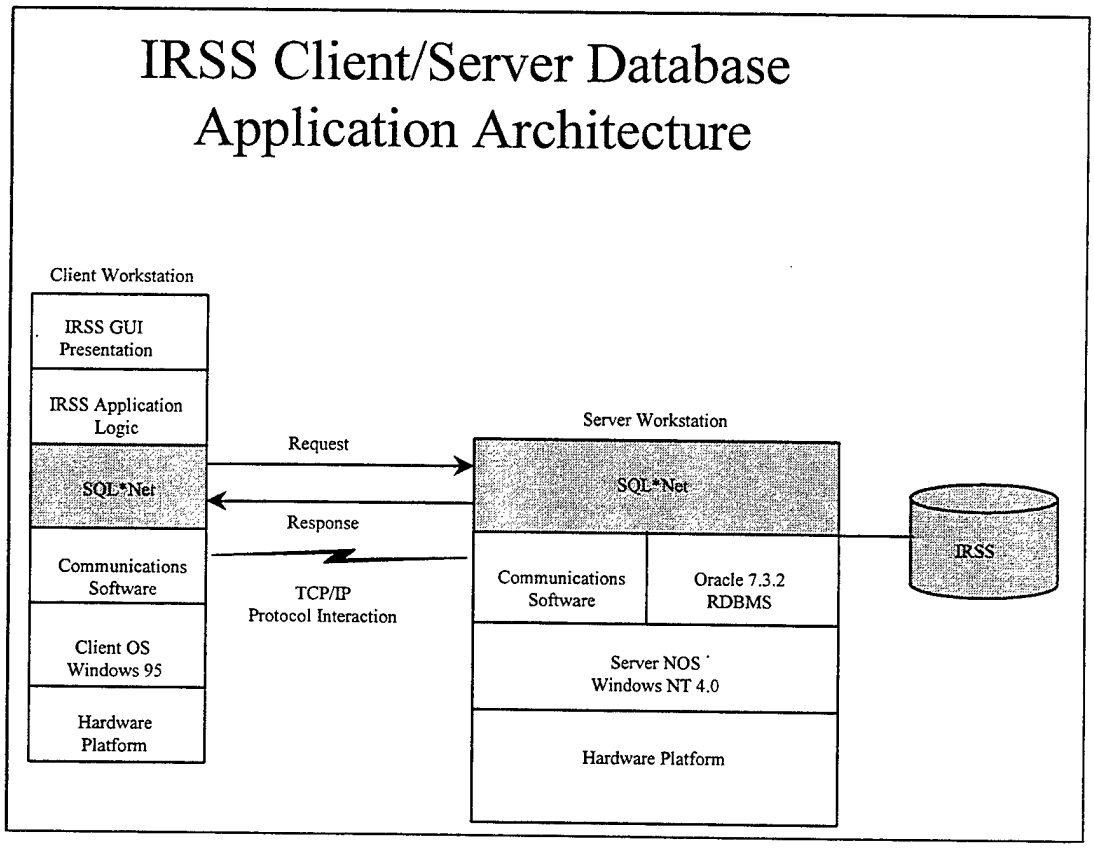
The diagram below depicts the replication scheme.



4.6 Connectivity

IRSS business processing is mostly client-based. Virtually all processing is done at the client except when interfacing with any of the features that require coordination, data retrieval or replication. These activities are performed by Oracle in the form of stored procedures. This architecture is probably the most common client/server approach in current use. The main advantage of this approach is to take advantage of the Pentium based processing power by offloading application processing to the desktop, while relieving the server to do more important functions such as replication and coordination. This approach minimizes network traffic and decreases the potential for bottlenecks or deadlock.

The diagram below illustrates the concept of splitting application logic between the client and server into the most common family of client/server applications; those that make use of RDBMS. In this environment, the server is a database server. Interaction between client and server is in the form of transactions in which the client makes a database request and receives a database response. However, all of the application logic (IRSS client software) is on the client side, while the server is only concerned with managing the database. SQL*Net provides the means for the client to talk to the server and uses the TCP/IP protocol over the Internet to carry the transmission.



4.7 Stored Procedures

In each IRSS database there are Oracle stored procedures that perform administrative functions instead of operational functions. These procedures are listed below:

- COORDINATE_ALL – This procedure queues all coordination related top-level Oracle stored procedures (PUSH_XXXXX) for execution at regular time interval.
- JOB_SUBMISSIONS_ALL – This procedure queues all replication related top-level Oracle stored procedures (PULL_XXXXX) for execution at regular time interval.
- JOBS_SCHEDULED – This procedure displays a list of queued database jobs for replication and coordination.
- JOBS_TO_KILL – This procedure is used to kill individual queued database job.
- KILL_ALL_JOBS – This procedure kills all queued database jobs related to IRSS.
- TIME_TO_CHANGE – This procedure resets the time interval in which coordination or replication executes.

4.8 To-Be Architecture (Versions 4 & 5)

This section discusses changes and modifications that are designed and currently being implemented in Versions 4 and 5 of IRSS.

4.8.1 Replication

Unlike previous version of IRSS, IRSS 4 and beyond will perform a complete refresh of its snapshot tables. Previous versions utilized a simple snapshot, using snapshot logs containing only transaction data from the previous refresh. With the implementation of N-server, snapshot logs will no longer be used and complete refreshes will need to be accomplished. This is a potential drawback of moving to the N-server architecture, but will outweigh the minor overhead incurred with using complex snapshots. A single snapshot table will be shared among all IRSS sites for a given master table. This process will increase the time required for snapshot table refresh, but the added overhead (scripts to maintain) will be minimal unless the amount of data contained in master table is substantial.

In IRSS version 4, following database tables are replicated to one site from all other sites: BASE, IPT, IPT_MEETING, MAJCOM, NOTIFICATION, PEOPLE, PEOPLE_IPT, SERVICE, ORGANIZATION, and UNIT.

4.8.2 Coordination

Contrary to replication, coordination is achieved by copying local data to other remote database sites. In IRSS, this process is called 'PUSH' process. When a user coordinates a task, data in the local database is copied to an associated temporary holding table. Then the data in

this holding table is propagated to tables at other IRSS database sites. The data will be propagated to other IRSS database sites every two hours.

Currently, data in tables COMMENT, DOCUMENT, RCM, and SECTION are copied onto holding tables COMMENT_COORDINATION, DOCUMENT_COORDINATION, RCM_COORDINATION, and SECTION_COORDINATION, respectively, for propagation to each IRSS sites. As the data is propagated successfully, the copied data in holding tables is deleted in preparation for the subsequent propagation. The PUSH process is performed by executing a series of custom Oracle stored procedures. The following top-level procedures are used in this coordination process:

- PUSH_INITIATE – The top-level procedure that initiates data propagation to all IRSS sites.
- PUSH_REMOTE – This procedure initiates data propagation for a single site.
- CREATE_DB_LINK – This procedure creates Oracle database link dynamically.
- DROP_DB_LINK – This procedure drops Oracle database link dynamically.
- CHECK_REMOTE_DB_CONNECT – This procedure tests for proper network connection between IRSS databases prior to data transmission.
- COPY_TO_DOCUMENT_REMOTE – This procedure pushes data in local DOCUMENT_COORDINATION table to remote DOCUMENT table.
- COPY_TO_SECTION_REMOTE – This procedure pushes data in local SECTION_COORDINATION table to remote SECTION table.
- COPY_TO_RCM_REMOTE – This procedure pushes data in local RCM_COORDINATION table to remote RCM table.
- COPY_TO_COMMENT_REMOTE – This procedure pushes data in local COMMENT_COORDINATION table to remote COMMENT table.

4.9 Certification and Accreditation (C&A)

Overview. We are currently investigating the C&A process and how it will affect IRSS and overall system performance. There will be a separate technical report addressing the C&A process, firewalls and security.

Requirement. We take for granted the information systems we use will always function properly. We assume the information we process on that system will never be disclosed to individuals who lack the clearance, authorization, or need-to-know. How do we know we won't be denied service, that the information on our system won't be deleted, or worse yet, unknowingly altered? The answer lies in the C&A process. By using a prescribed set of procedures and judgments, C&A gives a particular system assurance and authorization to operate securely in the targeted operational environment. There are six major components of the C&A process architecture. They are certification, accreditation, system security authorization agreement (SSAA), security testing and evaluation (ST&E), designated approving authority

(DAA), and certifying official.

Certification. A comprehensive, fully documented, evaluation of the technical and non-technical security features of an information system and other safeguards made in support of the accreditation process. When the documented level of protection and/or risk is considered to be acceptable by the DAA, system accreditation can take place.

Accreditation. Formal declaration by a DAA that an information system is approved to operate in a particular security mode using a prescribed set of safeguards at an acceptable level of risk.

SSAA. The SSAA is the depository for evidence showing the system meets the system security policy, all certification tasks are properly completed, the system is approved to operate, and a plan for maintaining the security posture/accreditation exists. Some of its contents include the system security policy, risk assessment report, C&A plan, and the system architecture. Attachments to the SSAA provide evidence the system security policy is properly implemented.

ST&E. Examines and analyzes the security safeguards of an information system as they are applied in an operational environment to determine the adequacy in implementing the system security policy. ST&I. provides evidence of a DAA's intention to comply with all appropriate laws, directives, and policies. It provides a measurement of a system s implementation of the system security policy.

DAA. The DAA is the individual who formally accepts security responsibility for system operation and officially declares it will provide an appropriate level-of-protection against compromise, destruction, or unauthorized modification under stated parameters of the accreditation. The DAA should be in the operational chain of command of the organization where the system is operating and is most affected by its failure/compromise.

Certifying Official. The Certifying Official is the individual responsible for making a technical judgment of the system's compliance with the system security policy objectives. They identify and assess the risks associated with operating the system. The Certifying Official has the responsibility for coordinating the various certification tasks and merging all the pieces of the certification package that will be presented to the DAA.

4.9.1 C&A Approach

Phase One - Pre-Certification. The primary goals of this phase are to gain an understanding of the requirements, translate the requirements into a system security policy, and to develop a plan of action to certify the extent to which the system implements the security policy.

Phase Two – Certification. This phase involves two activities: perform system analysis, and report findings/recommendations. Performing system analysis involves analyzing the security aspects of the system through the use of Security Testing and Evaluation (ST&E). It determines how well the security policy is employed throughout the system. Reporting

findings/recommendations involves documenting the results of the system analysis and generating the SSAA.

Phase Three – Accreditation. The DAA makes the accreditation decision after a careful review of the SSAA and the Certifying Official's recommendation. Accreditation allows the information system to operate within certain constraints and a defined environment with an appropriate level of protection.

Phase Four - Post Accreditation. This phase maintains the system security posture and accreditation. Post-accreditation tasks involve periodic assessments, active Information Protection (IP) involvement in system and/or operating environment changes, and continuous review of new threats and vulnerabilities.

Security and IRSS. In almost every case, the choices available to the administrator come down to a trade off between increased security at the expense of performance or administrative costs. IRSS is no exception to this rule. As we move towards a more secure environment , trade-offs will need to be assessed by the DAA in light of IRSS and other base-level systems. The ultimate decision will need to maximize throughput while minimizing security threats. System performance will be assessed at every juncture to ensure IRSS and the local security measures have the required protective measures in-place, in order to avert and prevent system penetration.

4.9.2 Database Security

In today's increasingly connected business environment, corporations are rushing to extend the availability of important information to mobile workers. The desire for anytime, anywhere access must be tempered with a proper appreciation of the vulnerabilities created when a user is not physically present. The problem of security can be broadly divided into two areas:

- Assuring that only legitimate users can access a system, and
- Making sure data is neither looked at nor tampered with as it travels across the network.

Security in IRSS is currently only userid/password to gain access to all the tables. Table access is controlled by the use of public synonyms, in other words, the IRSS user owns all of the tables but the synonyms or aliases allow other users to access them. Certain privileges to these tables are controlled by Oracle roles. For example, only Administrators can update the user_information table by adding a new user to the system.

The above security was all that was necessary during the prototyping stage of IRSS. The next step is to improve protection from using other applications such as a SQL editor. This type of protection will enhance the data security by not allowing any user to connect to the database outside of the IRSS application. That is to say, if you are not using IRSS to access the data then you will not even be allowed to login to the database.

Enhanced security will be implemented by removing all public synonyms and all rights to tables for all users. When the application attaches to the database (by use of a special password) it will only have rights to a table that defines roles for each user. Based on the userid the application will then assign roles (rights) to the current user. Upon closing the application all rights given to that user will be removed, there by rendering the database off limits to any other application or outside probing by that user.

4.10 Firewall Issues

Today, an unprecedented amount of efficiency and productivity are demanded of our armed forces organizations. In order to meet these demands, many organizations depend on groupware to collaborate across loosely coupled heterogeneous computer networks, namely the Internet and extranet, for quick and easy access to information throughout the world. The rapid advancement of services available over the World Wide Web (WWW) and e-mail along with other interactive groupware such as IRSS offer tremendous opportunities for bringing information to the desktop. As with any enhancements or improvements, with opportunities come risks.

Connecting an internal network to the rest of the world increases the possibility of unauthorized users gaining free access to an organization's sensitive data, possibly damaging files, stealing information, or compromising the national security. While attempts have been made to minimize the security risks inherent in connecting to external data networks, the majority of these attempts have fallen far short of providing complete network security. With recent releases and easy accesses to network security probing tools such as SATAN, COPS, Tripwire, Internet Security Scanner, and Cinco's Net-X ray and Web-X ray, once ordinary attackers have become very sophisticated and knowledgeable and are often able to breach the most complex security systems. Fortunately, numerous computer security firms have developed a group of software that would negate these new un-welcomed computer security threats while allowing legitimate computer users to conduct their daily assignments without disturbance.

4.10.1 Firewalls

Firewalls are secure gateways which control traffic into and out of the company's internal network. They are a generally a combination of hardware and software. There are basically two different implementation approaches employed by leading firewall vendors:

IP Filtering. This approach operates by blocking or allowing communication between networks or specific machines based solely on information contained in IP packet headers. While all firewall servers filter by IP address, only the firewall servers configured for stateful inspection filter by communication port numbers; when a firewall server does not restrict IP packet based on communication port number, it is said to perform stateless inspection. Packet filters are typically implemented in routers. They are configured using complex tables to indicate what communications protocols are allowed in or out of a particular network. Packet filters drop, reject, or permit packets based on destination IP address, source IP address, and application port numbers.

Application Proxy. In this approach, information flows through the firewall using an appropriate application proxy server, outside or header packets do not. Direct communication between the inside and the outside is severed. And the gateway acts as a data relay between inside and outside hosts, as defined by the security policy. Application or proxy firewalls are the most secure form of firewall. They run a small number of programs—called proxies—that can be secured and trusted. All incoming traffic is funneled to the appropriate proxy gateway for mail, HTTP, FTP and so on. The proxies then transfer the incoming information to the internal network, based on access rights of the individual user. Because the proxy is an application, it makes its decisions based on context, authorization, and authentication rules instead of IP addresses. This means that the firewall operates at the highest level of the protocol stack. This lets you implement security procedures based on a richer set of defensive measures. Proxies are relays between the Internet and the private network. The proxy's firewall address is the only one visible to the outside world. Consequently, the IP addresses on your internal network are totally invisible to the outside world. This is the preferred solution for implementing firewall security and is discussed later in our SQL*Net Proxy section.

4.10.2 Oracle SQL*Net

SQL*Net is Oracle Corporation's remote data access software. It enables both client/server and server/server communication across any network. Using SQL*Net, databases and their applications can reside on different computers and communicate as peer application.

SQL*Net establishes a connection to a database when a client or database server process acting as a client requests a database session. This can be the result of a request from a user or as the product of an automatically scheduled or data-triggered replication job.

There are three components in every SQL*Net connection:

- **Client** - Software that initiates the connection, whether client applications such as SQL*Plus, PowerBuilder clients, or an Oracle7 server (acting as a client).
- **TNS Listener** - Oracle-defined transparent network substrate (TNS); the packet protocol used by SQL*Net. The TNS Listener is software that establishes listen endpoints within a machine. These listen endpoints, IP ports in the case of TCP/IP, are well-known addresses that clients and servers use to initiate connections to the database.
- **Server** - Software that serves the client requests, such as an Oracle7 dedicated server process, or in the case of Oracle7's multithreaded servers, a dispatcher process.

Typically, the client calls a server by issuing a connection request to the listener process. The connection request addresses the target machine IP address, listener port number, and database instance identifier (SID). The listener accepts the incoming connection request and passes the connection to the appropriate server which then serves the client requests for the duration of a database session.

SQL*Net connections fall into two categories:

- Connections to a dedicated server
- Connections to a multi-threaded server

4.10.3 Dedicated Server

Dedicated server connections are connections in which one server process (or operating system process) is created for each database session initiated by a client. The server process is dedicated to servicing a single client's database requests. Typically, these dedicated servers are spawned at connect time; however, it is possible to configure a bank of pre-spawned servers in anticipation of connections.

After startup, the listener process listens for server connection requests on a well-known port. To create a connection, a client calls the listener on a well-known port. The listener receives a request and determines if the client is allowed to connect. If the listener denies the connect request, it returns an error to the client and continues listening for new connections. If the connection is allowed, the listener spawns a server process. Depending on operating system and TCP/IP implementation, the client connection is either bequeathed or re-directed to the server process so that client and server can communicate. The server process then performs authorization of the client and the listener continues to listen for new connections.

4.10.4 Multi-Threaded Server

A multi-threaded server (MTS) supports multiple database sessions per server process (operating system process). Using MTS reduces total system memory requirements and allows systems to support more database users with available memory. The listener process relays client connections to dispatcher processes, which in turn pass requests to MTS processes and return results to the client.

After startup, the listener process runs on a well-known port. MTS and Dispatcher processes are spawned. Dispatchers perform a wild-card listen and report the address issued back to the listener process. A connection is initiated when a client calls the listener on a well-known port. The listener receives requests and determines if the client is allowed to connect. If the listener denies the connect request, it returns an error to the client and continues to listen for new connections. If the connection is allowed, the listener redirects the client connection to the appropriate dispatcher for that client's protocol. The listener resumes listening for incoming connections.

4.10.5 Problems between firewall and Oracle SQL*Net

In theory, it is possible under certain conditions to configure SQL*Net to pass through a firewall. The ability to do this depends on the nature of the firewall itself (not all firewalls support this), the configuration of the server, and limitations of the operating system.

Firewalls which employ packet filtering may in general be configured to allow SQL*Net

traffic. Packet filters operate by blocking or allowing communication between machines or networks based on information contained in the IP packet headers. This information includes client and server IP addresses, and destination IP port number. Note that packet filters themselves don't offer much security. In order to minimize the security risk to your server, configuring a packet filter to allow SQL*Net traffic should only be done if you minimize the 'hole' in the firewall that you are opening up. Ideally, you would want to restrict incoming connections to a small number of named ports. For example, you might use one SQL*Net listener only, listening on port 1521. Note that unless your firewall understands SQL*Net and can verify that the connection coming through to port 1521 is really SQL*Net, you are always taking a chance that the hole through your firewall may be co-opted and used for something other than SQL*Net.

In some server configurations and some operating systems, you cannot easily limit port access in this manner. Systems running multi-threaded servers, pre-spawned servers, or ones which do not support port sharing require port redirection; that is, while the incoming connection is attempted at port XXXX, for example, the port 'redirects' the incoming connection to a different port number, say YYYY. The 'redirected' port number may not be known in advance, meaning that in order to allow this type of connection, you'd have to open up the range of ports to which the connection could potentially be redirected. Opening multiple holes in a firewall gives your firewall the consistency of 'Swiss cheese': lots of holes, meaning lots of potential security breaches.

4.10.6 Feasible solutions

This section discusses potential solutions to the firewall issues and concerns of AFOTEC and AFSPC.

4.10.6.1 The SQL*Net Proxy

The SQL*Net proxy is an application-level proxy that provides configurable access control and logging on the firewall gateway server, while protecting database servers from public access. The proxy is capable of passing Oracle's SQL*Net Version 2 or higher protocols between Oracle clients and servers. The SQL*Net proxy is based on the Oracle Multiple Protocol Interchange (MPI) .

The SQL*Net proxy passes Oracle SQL*Net requests through the firewall based on rules supplied by the firewall administrator. Rules can be based on a client's IP address or host name, the port number to be used on the firewall, a server's IP address or host name, and the database service identifier. With these rules, a company can configure the firewall to allow only specific Oracle clients to access an Oracle server through the firewall using the SQL*Net protocol.

The SQL*Net proxy-enabled firewall runs the SQL*Net proxy as a process and listens for requests on one or more configurable ports. When the firewall receives a request for service on this port, the proxy examines the request and checks configuration information to determine whether the client or host has permission to request the database service. If the host does not

have permission, the SQL*Net proxy logs the connection attempt and sends a refusal message back to the client.

If the client has permission to access the database service, the proxy makes a connection to the server on behalf of the client. When the connection to the server is made, the proxy logs the connection and passes requests to the server on behalf of the client. The SQL*Net proxy maintains the connection between the client and server until either side closes the connection or one of the connection timers expires.

Unfortunately, SQL*Net proxy is not available directly from Oracle to customers. The functionality is provided by firewall vendors which have integrated the new technology into their current product offerings. By purchasing a firewall solution from a vendor who has integrated the proxy, Oracle customers will get this functionality.

Another reason the SQL*Net application proxy is not directly available to customers is because Oracle corporation is licensing source code for the SQL*Net application proxy, as well as pieces of SQL*Net and the kernel, to firewall vendors so that they may integrate the proxy into their firewalls. Oracle only provides source code under a strict licensing agreement and Oracle will not provide source code to customers.

The firewall software/hardware products and vendors that currently support Oracle SQL*Net are listed below:

- Checkpoint Software Technology - FireWall-1 version 3.0
  http://www.checkpoint.com/products/firewall-1/descriptions/products.html
- Cisco-Global Internet - Cisco PIX Firewall version 4.1
  http://www.cisco.com/warp/public/751/pix/index.shtml
- Digital Equipment Corporation – DEC - AltaVista Firewall97
  http://www.altavista.software.digital.com/firewall/index.asp
- Milkyway Networks - SecureIT Firewall for NT version 4.0.3 (Supports ODBC-compliant statement only. Limited.), SecureIT Firewall for Solaris version 4.0
  www.milkyway.com
- Raptor Systems - EagleNT version 5.0, EagleUnix version 5.0
  http://www.raptor.com/products/ds/eagle/eagle.html
- Secure Computing - Secure Computing Firewall for NT version 3.0.1, Sidewinder Security Server version 3.2, SecureZone version 1.0 http://www.sctc.com/products.html
- Trusted Information Systems - Gauntlet Internet Firewall latest version – no version number available. http://www.tis.com/prodserv/gauntlet/index.html

4.10.6.2 IRSS specific firewall issues known to date.

Issue 1 - IP Address Validation. The client has expressed the desire to eliminate the time-consuming process of tracking each IRSS client machine's IP Address (of which there are

hundreds) in order to keep the list of authorized IP Addresses current; the popular accepted use of Dynamic Host Configuration Protocol (DHCP), server aggravates this problem. A DHCP server can supply IP addresses and configuration settings automatically to client machines. The best way to implement this is to ensure that all communication attempts to Oracle Database Servers are done under one IP Address. The ideal situation is for all communication attempts to go through the Oracle Database Server. Currently, communication attempts are made from the Oracle Database Server whose IP Address is authorized, and from each IRSS client machine whose IP Addresses are not authorized. Therefore, any communication from an IRSS client machine results in the Firewall rejecting the attempt. Below are possible solutions that have been identified for this problem.

Solution 1 to Issue 1 – Use of a Proxy Server (Firewall Server)
Implement a Proxy Server so that all communication from the Oracle Database Server and from each IRSS Client machine goes through the Proxy Server. By doing this, only one IP Address (the Proxy Server's IP Address) must pass validation through the firewall. The Proxy Server's IP Address must be included in the firewall's list of authorized IP Addresses. In fact, most firewall servers are proxy server by nature, and thus all outgoing traffic may be directed to the site's proxy-configured firewall. However, directing network traffic through a firewall server reduces network throughput. See Diagram on next page.

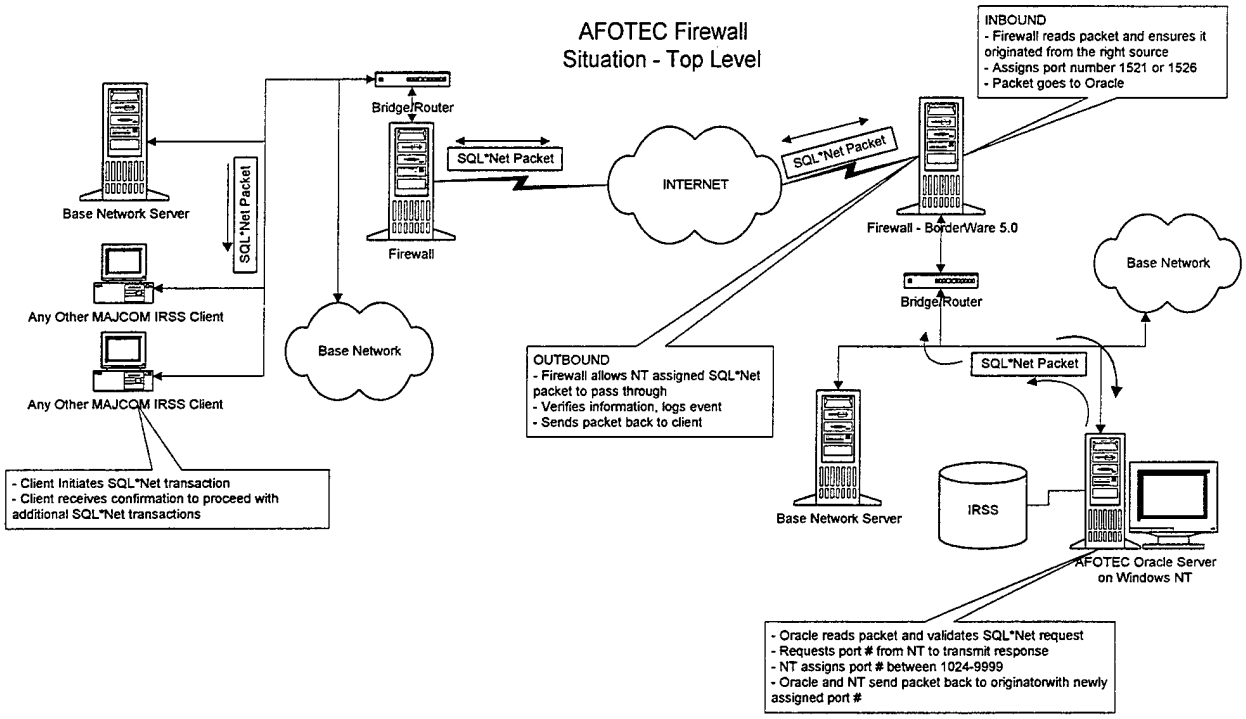
Solution 2 to Issue 1 – Upgrade to Oracle 8 and Implement Code Changes. Currently, the IRSS database has been implemented using Oracle Workgroup Server version 7.3. When moving a binary data, such as a bitmap image, between IRSS databases, the data is moved from one database server to a client machine, then it is moved from the client machine to the destination database server instead of moving the data from one server to another server directly. This is due to the current version of Oracle's inability to manipulate binary data size in a standard SQL statement.

Upgrading to Oracle 8 may successfully allow remote procedure calls to be made from the IRSS client machine to the Oracle database server. This will force any communication attempt from an IRSS client machine to go through the Oracle database server. By doing this, only one IP address, the Oracle database server's IP address, may pass through the firewall's validation. The Oracle database server's IP address must be included in the firewall's list of authorized IP addresses. However, this will require extensive changes to PowerBuilder codes in the IRSS client software and possibly adversely impact some firewall solutions (in the near term)

Issue 2 -Communication Port Validation. Once the source IP address has been validated by the firewall, the communication connectivity is authorized via well-known listener communication port. Because Windows NT is a multi-threaded server and the Oracle RDBMS for Windows NT is a multi-threaded database server, once the client request is validated by Oracle listener process, the client to server communication is redirected to a new communication port, usually between 1024 and 9999. Apparently, this new port number is selected in near random manner; NT allocates to Oracle the lowest unused port number above 1023, which the firewall is not able to predict beforehand. Therefore, it is not possible to know in advance what communication port must be added to the firewall's list of authorized communication ports. The random creation of the new communication port results in the firewall rejecting the subsequent

client requests directed to the new communication port number because it is not found in the firewall's static authorization list. Below are possible solutions that have been identified for this problem.

Solution to Issue 2 – Use an Oracle SQL*Net Compliant Firewall. Using an Oracle SQL*Net compliant firewall listed previously will eliminate this problem. Currently, AFOTEC is experiencing difficulty in enabling their firewall to process SQL*Net traffic in a manner that meets the security needs of their network manager. They are using BorderWare version 5.0 with patches 1 through 7 installed. Patch 5 enables SQL*Net 2.3 to be supported by BorderWare 5.0. This Software Update includes a new proxy the supports Oracle SQL*Net connects on ports 1521 and 1526. This proxy allows all inbound and outbound configurations. The patch depends on patch 3 to be installed and has no exclusions nor conflicts with any previous patches.
See Diagram below.



APPENDIX A
IRSS REQUIREMENTS TABLE

APPENDIX A – IRSS Requirements Table

ID	Functional Requirement
1	Track Mission Needs
2	Track mission deficiencies and solutions and provide justification.
3	Identify and track program issues.
4	Generate Pre-formatted Report Documents (Fact Sheets, Integrated Priority Lists.)
5	Provide user login security (e.g., login and password).
6	Track needs being satisfied by acquisition programs.
7	Provide query and ad-hoc reporting access.
8	Support assignment of user profiles, and access control privileges (e.g., read only, edit, add, etc.).
9	Support multiple administrators for various levels and modules: (e.g., System Administrator for specific functional modules) Dependent upon the size of the installation site, may want to have the capability to assign separate administrators for different parts of the database (e.g., requirements tracking, reference data used by the application - allowed values and picklist items, financial, etc.).
10	Provide the Mission Area Plan source for the need.
11	Provide description of the deficiency/need.
12	Track OPR (office) and Action Officer assigned to Needs - to - Solution projects and information product generation.
13	The system should support sets of requirements.
14	Support the identification of key performance parameters.
15	Allow comments (notes) related to need.
16	Provide capability to calculate schedule of project milestones to be used for process tracking, control, management, and coordination:
17	View summary of all comments, and provide reporting capabilities for comments and resolution data.
18	Filter comments so that specific comment categories (e.g., critical) can be viewed together.
19	Track personnel information related to Action Officers and Points of Contact (e.g., name, rank, title, organization, address, e-mail, phone/fax, etc.). Include the capability to produce personnel rosters with this data.
20	Aid user in assigning unique identifiers (document and project identifier), attributes (title, document type), and key tracking data for documents and other information products).
21	Correlate supporting information elements associated with needs - to - solutions. It should follow that correlation also relates to work with other processes logically following the requirements generation system. (acquisition, PPBS, implementing/scheduling, program management, etc.) Data hook is threshold.
22	Have capability to consolidate received comments.
23	Maintain versions of requirements and hierarchies.
24	Each requirement (need) should/must be individually identifiable and able to be traceable.

ID	Functional Requirement
25	Ensure each requirement must be associated with a parent requirement or mission need.
26	Handle data maintenance/management.
27	Manage processes for Data Entry and Updates.
28	Track tasks and staff actions associated with the development of an organizations projects: needs, requirements, and solutions.
29	Track approval levels, POC, OPR, etc. associated with needs - to - solutions.
30	Capture Operational needs data.
31	Provide requirements history/rationale.
32	Establish project (must assign it to an organization and assign resources (e.g., POC).
33	Track funding decisions of requirements review panels.
34	Track tasks, staff actions, suspense's associated with the organization's needs - to - solutions. Includes when validated and approved, by whom (organization).
35	Track dates of reviews, briefings, prioritization, funding approval. Track results of briefings and review (i.e., prioritization and rankings from review boards).
36	Annotate comment categorization, and include capability to provide to the Action officers a count by project, of what category type: critical, significant, administrative, comments for a particular information product (e.g., document) that requires resolution.
37	Allow comments to be linked to specific sections of draft documents and information products.
38	Provide application interface capability for IRSS to incorporate other current software tools and those under development. Incorporate an open-design strategy as part of IRSS development.
39	Record all deficiencies and needs without regard to material / non-material solutions.
40	Account for origin of requirements and needs. Link needs and the hierarchy of requirements to other referenced source documents where requirements are drawn from.
41	Track Approval of requirements and rationale statements.
42	Track which projects, Needs, Capabilities, Requirements, Information Products etc. that each agent is the author of. Also include each user that has modified these information elements.
43	Establish and maintain accountability and traceability.
44	Generate fact sheets document (summary of project/document database).
45	Database flexibility - Provide for flexibility to account for future change / modifications to the database.
46	Provide PC platform compatibility, with applications transportable on notebook computers.
47	Review tracked data records for discrepancies (such as missing key dates, missing info, data problems, missing executive summary).
48	Data Currency (when was the data last updated).

ID	Functional Requirement
49	Control access down to application module, record level and data fields on records (where applicable).
50	IRSS needs to be scaleable for use from stand alone operations to multi-user operations with local and wide-area accessibility, and multi-group collaboration.
51	Provide for database tailorability and extendibility, not involving structural and entity relationship changes. (e.g. allowed values, modify picklists, etc.).
52	Identify needs that have joint/other command applicability.
53	Provide the priority of mission needs to solution.
54	Identify IPT members and associated organizations (down to individual personnel).
55	Capture supporting documents associated with mission needs.
56	Correlate Needs-To-Solutions with Mission Area Plan/Roadmap deficiencies, Intelligence Support Plan deficiencies, Inspector General, Staff Assistance Visit, Joint Uniform Lessons Learned findings and recommendations, mishap investigation recommendations, current weapon system deficiencies, and other sources.
57	Track whether a physical, signed, legal, official copy of a document exists and where it is located.
58	Track resources (Personnel/Responsible Organizations) to projects, Needs-To-Solutions information elements and other processes that are managed and tracked:
59	Define needs for primary and secondary mission areas.
60	Provide impact of the need (training, Intel).
61	Classification markings of data and replication.
62	Print reports to text files.
63	Executive Level Summary Report
64	Share document/database information digitally via e-mail and make available via Internet Web Site.
65	Generate Need Documents (MNS/CMNS).
66	Generate requirements documents (ORD, CRD, CSRD). Include automated capability to generate RCM (Part 1, 2 and 3) with ORDs.
67	Provide the capability to import requirements sets from other tools and sources.
68	Provide an "executive - Level Views" of the data to show management a "virtual oversight" model to aid them in navigating the IRP knowledge base at various levels for analysis, review and approval actions, process and management control, research, etc.
69	Provide generic query capability.
70	Describe the required capability that generated the need.
71	Provide access to and maintain database repository.
72	Capture solution data from the MAP process.
73	Provide External interfaces to posted information, including access control for managing external views into the information and process.
74	Address human systems interface issues, with focus on "user seductive", keeping the interface simple and clean.
75	Track threshold and objective values as they evolve during the requirements definition process.

ID	Functional Requirement
76	Provide ability to identify widows and orphans.
77	Track cost, schedule, and performance data.
78	Document and track "exit - Criteria" (Success factors for project to successfully proceed to next step).
79	We need multiple ways to access IRSS information (e.g., dial-up access and Internet access, depending upon your individual capabilities.).
80	Have capability to view links and hooks to references (e.g., requirements, analysis, and justification sources) associated with information products.
81	Support approval actions at various levels.
82	Access Needs-to Solutions data for all correlated and flagged items.
83	Must have an electronic linkage back into the mission area plan. If possible, the linkage should go back as far as the S-T-T in the MAA. This linkage must serve as the source for the traceability of the deficiency (need) all the way into the acquisition process. this should include the entire strategy to task process and all of the sub functions of the strategic planning process. Each individual requirements (need) must be able to be linked and tracked. Also, the traceability as on requirement changes based upon mission changes in S-T-T must be tracked, rational identified and the cascading effect of the change seen (e.g. impact to other requirements).
84	Provide a visual way to identify that a comment exists on a portion of a document.
85	Version control on mission needs.
86	Provide multiple ways to access the IRSS from dial-up access and Internet access, depending upon your individual capabilities.
87	Track association of documents to acquisition program.
88	Provide a checklist of the processes steps (process aid).
89	Provide access to and maintain document archive.
90	Track what reviewers have been provided information products for review, and review status.
91	Handle review, comment, and comment resolution in near-real-time.
92	Enable the use of graphics.
93	Provide capability to execute information product review, comment, and comment resolution activities within an automated environment (over networks and e-mail). Needs to do this in real-time in order to have an impact on the work/rework cycle time.
94	Provide external interfaces. Provide external interfaces remote users with an need for information from the requirements process to view posted information and process data.
95	Summary and detailed information views supported by the following capabilities.
96	Execute coordination tasks and staff actions associated with the organizations.
97	Provide IRSS software context-sensitive help.

ID	Functional Requirement
98	Define program related data that characterizes solutions resulting from requirements process. Provides traceability from Needs through Requirement definition steps tracking the resulting development of both non-material and material solutions.
99	Correlate requirements and related information products to specific Defense Systems.
100	Track development of information products associated with Needs - to - Solution projects.
101	Automate dissemination aspects of information product staffing (drafting, coordination, distribution, etc.), and subsequent electronic dissemination of all phases/versions of the document across the requirements community.
102	Comments made on a requirement should be fed back to the "system" and available to the AO to begin the resolution as soon as possible. (e.g., daily).
103	Implement control to have specific individuals responsible for reviewing consolidated comments.
104	Provide source and amount of funding associated with projects and Needs-to-Solutions. Also, include a breakdown of funding over time.
105	Definition of solutions in IRSS should track development of solutions through modernization or acquisition process.
106	Insert individual statements (document information elements) and groups of related statements into document (e.g. when a parent statement is inserted, automatically bring in the children statements.).
107	Provide on-line IRSS software operation tutorials.
108	Drag and drop capability to create links and hierarchies.
109	Data Concurrency.
110	Provide capability to document partial/interim responses.
111	Provide capability to execute coordination, staffing and review activities within an automated environment.
112	Provide on-line process tutorials: high level view of the requirements process.
113	Provide a summary view of different information classes to assist user in rapidly screening larger groups of data.
114	Automate the generation of the RCM.
115	Ability to import and export files to support use of electronic group support systems.
116	Display and present process for building a requirement document from direction/start to draft ready signature to go AF wide comment.
117	ABIDES financial data.
118	Link need to Primary and Secondary Mission Areas.
119	Associate a Test with a Requirement.
120	Access to integrated library: initial with key DoD and AF directives and instructions related to requirements without links.
121	Provide graphical view of the process with an indication of where requirements are in the process. Provide the ability to see where a particular requirement is in the process and other filters to help digest the information in the graphical view.

ID	Functional Requirement
122	Generate Program Documents AF Form 1067.
123	Correlate with Linkages to PMD data and Review Board decisions.
124	Provide a completeness checker that examines a requirements set, including the defined relationships, and identifies requirements that are incomplete, requirement sets that have logical inconsistencies, and requirements which include specification errors.
125	Provide generic capability to identify and document requirement community issues for disconnects and resolution.
126	Bring in other documents and search them for source requirements;
127	Accommodate the need to link to SPO systems engineering environment and RFP generation tools.
128	Provide IRSS software operation wizards.
129	Develop a best-practices or corporate-knowledge help area to share lessons-learned, latest updates, answers to common questions, etc. with other IRSS participants.
130	Provide some privacy mechanisms so that comments can be shared with an IPT or closely held between the owner and the reviewer. This should be user definable.
131	Provide electronic dissemination of up to at least Secret-level requirements.
132	Identify word patterns for search criteria.
133	System needs to operate at least the Secret level (desire capability to work off-line and up to Top Secret/SAP/SAR).

APPENDIX B
Proof-of-Concept Test Results

APPENDIX B - Proof-of-Concept Test Results

The IRSS proof-of-concept test was conducted at WPAFB on 10 –11 December. The test team consisted of representatives from each MAJCOM, members of the BAH development team, and Air Force Research Laboratory personnel.

The technical architecture used to support the testing included eight workstations, two running Windows NT, and six running Windows 95. One of the NT workstations was also functioning as the AFRL IRSS database server. Each workstation was running the client IRSS software version 2.9b. This configuration was established to allow testers to work on their home, remote databases. The following list captures the testers and the workstations that were used for the test script execution:

1. Workstation 1 – HQ USAF/XORPD, Maj Moen
2. Workstation 2 – HQ AFSOC/DOXR, Maj Snyder and Lt Col Spence
3. Workstation 3 – HQ ACC/DR, Mr. Jim Zweigler and Mr. Lee Sink
4. Workstation 4 – HQ AFSPC/DR, Mr. Larry Rainey
5. Workstation 5 – AETC/XP, Capt Andy Gwinnup
6. Workstation 6 – AFMC and AFRL/HESS, Mr. Jim Roe and Lt. Christine Martinez
7. Workstation 7 – HQ AFSOC/DOXR, Capt Mundy and Mr. Glynn Cooper
8. Workstation 8 – AIA and HQ USAF/XOII, Mr. Bob Uhl and Capt Holder

Non-existent or poor connectivity forced most of the test participants to execute the test scenarios on the local AFRL server. HQ ACC/DR was the only organization to complete the majority of tests on their home database server. In order to perform testing in an environment that simulated real-life usage, BAH migrated a subset of ACC's IMPP data to ACC's IRSS database server. Due to data replication implications, ACC's communication with other IRSS database servers was temporarily suspended.

Prior to testing, each test team was given a copy of the test scenarios which they were advised to follow in a self-paced format. The testers were also directed to record all of their responses on the scenarios, which would be collected at the end of the training to compile the final results. Members of the BAH development team were present to answer any questions and respond to problems that may arise during testing.

The testing lasted approximately one and a half days, during which time the testers, and the BAH team, identified some intra-server communication and replication problems. The BAH technical team worked throughout the training to resolve any anomalies and subsequently released software version 2.9c.

Test Results
The Proof-of-Concept test team provided generally positive feedback throughout the testing execution.

A summary of each test team's evaluation of the functional test scenarios are provided in the following table:

TEST SCENARIO	AFMC	ACC	XORD	AIA	AFSOC	AFSPC	AFSOC	AETC	MEANSCORE
SYSTEM ADMINISTRATOR									
SA1: Add/Change Table Data	5	5	5	5	5	5	4	5	4.86
SA2: Establish Access Controls	3	4	5	5	1	5	5	5	4.29
ACTION OFFICER									
AO1: Deficiency Tracking	4	4	5	5	1	5	5	5	4.29
AO2: Project Creation	4	3	4	4	1	5	4	4	3.57
AO3: Draft MNS	5	4/5	3	4	1	5	4	5	3.67
AO4: Reconcile MNS Comment	5	4	4	5	1	NT	3	5	3.67
AO5: Draft ORD	5	4	4	3	1	NT	4	2	3.00
AO6: Reconcile ORD Comments	5	3	5	NT	1	NT	NT	NT	3.00
AO7: Generate Reports	NT	3.5	5	3	4	NT	2	3	3.42
MANAGER									
MGR1: Examine Document Reports	NT	4	5	5	4	NT	2	5	4.17
MGR2: Examine Project Reports	NT	4	5	4	4	NT	2	5	4.00
MGR3: Ad Hoc Query	NT	3	5	4	4	NT	4	5	4.17

NT – Not tested

An unexpected loss of network connectivity to the AFRL database server caused a delay in data replication and consequently impacted the test execution of the A06: Reconcile ORD comments scenario. Due to the delay, testers did not receive coordinated documents with sufficient time to execute the test script, which is reflected as "NT" in the above matrix.

After reviewing the consolidated test results, it appeared that AFSOC's results were inconsistent with the rest of the test team. AFSOC evaluated several test scenarios with a "1" ranking. However, in their test results AFSOC noted that the "1" ranking referred to the fact that the test scenarios did not address all of the requirements, NOT that the functions tested were incomplete.

Overall, users reported that they would be able to use IRSS in their day to day operations if they were given instructions on how to deploy the application within their organization. The testers

were enthusiastic about the software's potential, but generally agreed that some areas needed to be expanded or revisited altogether. The testers identified the following items as such:

- Deficiencies and Justifications – Users believed the module needs to be more clearly defined, and expanded to be suitable for use in the planning community
- Data Sharing/Security – The complexity of the security implementation proved cumbersome when reviewing and coordinating documents.

Additionally, ACC suggested that many of the requirements being tested would more suitably be addressed in an operational testing environment.

Evaluation Criteria

The Test participants are to evaluates each test scenario and records the results. Test scenarios consist of verifying functionality and then evaluating that functionality on a pass or fail basis.

Overall test scenario ranking follows a five-level evaluation scheme:

Level	*Description*
1	Item cannot be evaluated or evaluation is incomplete. This is normally the result of an incorrect test setup or an unsatisfied dependency. If this category is checked, the reason should be stated. If the test could not be evaluated because the test was inadequately specified, the test plan needs to be corrected.
2	Item does not pass, and an operational workaround for the problem does not exist.
3	Item does not pass, but an operational workaround for the problem exists.
4	Only cosmetic errors were found in evaluating the item.
5	No problems were found.
N/A:	This item was not tested at this site.